INPUT 64

DAS ELEKTRONISCHE MAGAZIN 6/85

Infos · News · Programme · Unterhaltung · Tips DM 12,80



Textadventure:
Fuchsjagd

Neuer Kurs:
Musik-Programmierung

Speicher gespart:
BASIC-Compactor

Statt Pauken:
Mathe mit Nico

News, 64er-Tips, Tools, Anwenderprogramme

Dokumentation und Bedienungshinweise

INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1



Unser Bestseller!

6. Auflage Ein BASIC-Buch auch für Nicht-Techniker, Nicht-Mathematiker, Nicht-Computer-Profis!

Siegmar Wittig

BASIC-Brevier

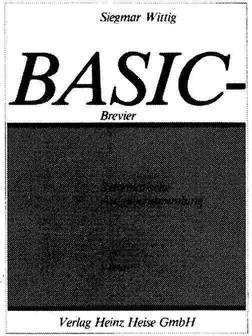
Eine Einführung in die Programmierung von Heimcomputern

6., erweiterte Auflage

Berücksichtigt speziell die BASIC-Versionen von Apple, Atari, Commodore (mit besonderen Hinweisen für VC-20 und C-64), Epson, Heath-Zenith, Tandy, Texas Instruments, Sinclair ZX81 und ZX Spectrum.

238 Seiten mit 15 Abbildungen, 6 Tabellen, zahlreichen Programmbeispielen, Programmieraufgaben mit Lösungen und einer Sammlung von 10 ausführlich beschriebenen Programmen. Format 18,5 x 24 cm. Kartoniert, DM 34,00. ISBN 3-922 705-01-4

Die ideale Ergänzung zu jedem BASIC-Lehrbuch, aber auch eine einzigartige Programmsammlung!



Siegmar Wittig

BASIC-Brevier. Systematische Aufgabensammlung.

207 BASIC-Aufgaben mit kommentierten Lösungen und zahlreichen Lösungsvarianten.

3. Auflage 1983. 210 Seiten. Format 18,5 x 24 cm.
Kartoniert, DM 29,80. ISBN 3-922 705-02-2

Diese Aufgabensammlung kann neben dem Lehrbuch **BASIC-Brevier — Eine Einführung in die Programmierung von Heimcomputern,** aber auch neben jedem anderen BASIC-Lehrbuch oder Hersteller-Handbuch verwendet werden. Die Lösungen sind in Microsoft-BASIC geschrieben.

Die Aufgabensammlung stellt aber auch für den fortgeschrittenen Programmierer eine einmalige Sammlung von wichtigen Programmsequenzen dar, denn sie enthält u. a. zahlreiche Programme zu den Bereichen Mischen, Trennen, Einfügen, Sammeln, Suchen und Sortieren von Daten, Konversionsmethoden, Simulation, Bit-Manipulation u.v.m.

Die Anordnung der Aufgaben ist systematisch. Zu allen wichtigen BASIC-Sprachelementen werden Aufgaben angeboten. Die Aufgaben werden zunehmend umfangreicher und schwieriger. Ihre Lösungsvorschläge enthalten mehr und mehr unterschiedliche Sprachelemente. Tabellen erlauben die Auswahl von Aufgaben, die mit bestimmten Sprachelementen oder Kombinationen davon gelöst werden.

Verlag Heinz Heise GmbH · Postfach 27 46 · 3000 Hannover 1

INHALTSVERZEICHNIS INPUT 64

6/85

Liebe 64er-Besitzer(in),

potentielle Aufsteiger können beruhigt sein. Wer sich mit dem Gedanken trägt, vom C 64 auf den Commodore C 128 umzusteigen, braucht auf INPUT 64 nicht zu verzichten.

Erste Tests der Redaktion auf der Hannover-Messe ergaben, daß nach dem "GO 64" (der Befehl zum Umschalten in den 64er-Modus) INPUT 64 problemlos auf dem 128er lief. Die Kompatibilität des C 128 zum C 64 scheint also gewährleistet zu sein. Endgültige Aussagen können wir natürlich erst nach ausgiebigen Tests machen. Demnächst also mehr zu diesem Thema. Wir bleiben am Ball.

Viel Spaß mit INPUT 64!

Hinweise zur Bedienung

1.) Entfernen Sie - bei ausgeschaltetem Rechner - eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD"INPUT 64",8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise ("Bitte Band zurückspulen" und so weiter ...).

4.) Merken Sie sich fünf Kommandos:

- (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

- (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert. Von Diskette wird automatisch neu geladen.)

- (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

- (CTRL) und (s) steht für SAVEn und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwendungsprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisungen - Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programmanfang möglich - entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

- (CTRL) und (b) ermöglicht einen Bildschirmausdruck - natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Voraussetzung ist Geräteadresse 4.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist.

Auf der Rückseite der Verpackung finden Sie das Inhaltsverzeichnis der Kassette/Diskette.

Auf einen Blick: INPUT 64-Betriebssystembefehle

Inhaltsverzeichnis aufrufen	CTRL und i
Hilfsseite aufrufen	CTRL und h
Programm abspeichern	CTRL und s
Bildschirmausdruck	CTRL und b
Titelbild abkürzen	CTRL und q

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner "abstürzen". Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

SID-Kurs

Der neue Kurs in INPUT 64 führt Sie in die Grundlagen der Musikprogrammierung und die wesentlichen Funktionen des Sound-Chip ein.

Als Hilfestellung für eigene Programmierversuche werden wir im Beiheft jeweils die wichtigsten besprochenen Adressen noch einmal aufführen:

FUNKTION		ADRESSE dez.	ADRESSE hex.
Sound-Chip Basisadresse		54272	$D400
Stimme 1 Tonhöhe	Low-Byte	54272	$D400
"	High-Byte	54273	$D401
Stimme 2 Tonhöhe	Low-Byte	54279	$D407
"	High-Byte	54280	$D408
Stimme 3 Tonhöhe	Low-Byte	54286	$D40E
"	High-Byte	54287	$D40F
Lautstärke alle Stimmen		54296 (Werte von 0 bis 15)	$D418

HILFSPROGRAMME 1: PRINT AT und INKEY

Wir bieten Ihnen diese beiden Routinen in zwei Versionen an. Diese Versionen unterscheiden sich nur in dem belegten Speicherbereich. Die SYNTAX ist in beiden Fällen (bis auf die unterschiedliche SYS-Adresse) identisch.

Für beide Versionen gilt, daß Sie das gesicherte Programm (natürlich nur außerhalb von INPUT 64) als normales BASIC-Progamm laden können.

Sie werden sicherlich diese Routinen auf Ihren Datenträger sichern wollen. Beachten Sie aber bitte, daß bei diesem Programm das Sichern nur an zwei Stellen möglich ist. Sie müssen vom Menue aus eine der beiden Versionen direkt anwählen. Das Demonstrations-Modul wird in beiden Fällen nicht mit abgespeichert

PRINT AT und INKEY am BASIC-Anfang:

Wenn Sie diese Version nach dem Laden von Ihrem Datenträger mit RUN starten, wird der BASIC-Anfang automatisch auf $0b00 "hochgelegt".

Sie sollten jetzt ein NEW eingeben, damit alle BASIC-Zeiger richtig gesetzt werden. Danach können Sie ein eigenes Programm (nach)laden oder auch im Direktmodus ein neues Programm schreiben.

Beachten Sie aber bitte, daß sie vor dem SAVEn den BASIC-Anfang mit POKE 44,8 wieder heruntersetzen müssen, da sonst die Routinen nicht mit gesichert werden.

Die SYS-Adressen sind: SYS 2691 für den PRINT AT - Befehl und
SYS 2096 für den INKEY - Befehl.

PRINT AT und INKEY im Bereich ab $cc00:

Auch diese Version wird mit RUN gestartet. Eine Verschieberoutine sorgt dafür, daß die vielen Bytes in den Speicherbereich ab $cc00 geschaufelt werden. Diese Verschieberoutine sorgt auch selbständig für ein NEW und gibt einen Hinweis auf dem Bildschirm aus.

Die SYS-Adressen sind: SYS 52224 für den PRINT AT - Befehl und
SYS 52256 für den INKEY - Befehl.

Diese "obere" Version kann nicht direkt an das eigene Programm angebunden werden, sondern muß jedesmal neu geladen werden.

Bei den folgenden Beispielen werden wir der Einfachheit halber nur die SYS-Adressen der "unteren" Version verwenden.

Beschreibung des PRINT AT-Befehls:

Mit diesem Befehl können Sie den CURSOR (und damit den Beginn der Ausgabe) auf einen bestimmten Zeilen- und Spaltenwert setzen.

Die SYNTAX des PRINT AT - Befehls ist:

SYS ADRESSE, ZEILE, SPALTE, TEXT

Alle Angaben können auch in Form von Variablen übergeben werden. Für die Zeilennummer können Werte von 0 bis 24 und für die Spaltennummer Werte von 0 bis 39 eingesetzt werden. (Zeile 0 und Spalte 0 befindet sich oben links auf dem Bildschirm) Für TEXT kann alles eingesetzt werden, was auch bei dem normalen PRINT-Befehl erlaubt ist.

Beispiel: sys 2691, 10, 15, "test"

Bei diesem Beispiel beginnt das Wort 'test' in Zeile 10 und Spalte 15. (Statt "test" könnte auch eine VARIABLE oder eine Kombination aus VARIABLE und TEXT stehen.)

Dieser SYS-Aufruf ist auch im DIREKTMODUS möglich.

Beschreibung des INKEY-Befehls:

Dieser Befehl beseitigt alle denkbaren "Unfertigkeiten" des INPUT-Befehls beim C 64. Sie können bestimmen, wo die Eingabe auf dem Bildschirm stattzufinden hat, wieviel Zeichen eingegeben werden dürfen und welche Zeichen überhaupt erlaubt sind.

Wenn wir Ihnen nun noch sagen, daß Sie - innerhalb des durch die Länge vorgegebenen Feldes - frei editieren können(CURSOR rechts/links, HOME, CLR-HOME, DEL und INST), und daß Sie noch optional ein Markierungs-FLAG setzen können, werden Sie uns sicherlich zustimmen, daß unser INKEY-Befehl kaum noch Wünsche offen läßt.

Die SYNTAX des INKEY - Befehls ist:

SYS ADRESSE,ZEILE,SPALTE,LÄNGE,DEFSTRING,ZIELSTRING (,FLAG)

Die Angaben für ZEILE und SPALTE entnehmen Sie bitte dem PRINT AT - Befehl. Mit der LÄNGE bestimmen Sie die maximale Anzahl der einzugebenen Zeichen. Achten Sie aber darauf, daß dieser Wert nicht größer als 40 minus SPALTE sein darf, da die Eingabe in einer Zeile stehen muß.
Der DEFSTRING enthält alle Zeichen, die bei der Eingabe erlaubt sein sollen. So ist es zum Beispiel möglich, eine rein numerische Eingabe durch "0123456789" festzulegen. Der ZIELSTRING (hier muß (!) eine STRINGVARIABLE eingesetzt werden) legt fest, welcher Variablen die Eingabe zugeordnet wird.

Im folgenden Beispiel gehen wir davon aus, daß das FLAG nicht gesetzt ist, das heißt, nach dem ZIELSTRING steht keine weitere Variable (auch kein Komma). In diesem Fall kann die Eingabe nur mit der RETURN-Taste beendet werden.

Beispiel: 100 sys 2096,5,10,20,"0123456789",i$

Bei diesem Beispiel beginnt die Eingabe in Zeile 5 und Spalte 10. Es werden nur Zahlen angenommen. Es ist ferner unmöglich, mehr als 20 Zahlen einzugeben. Wenn Sie sich bei der Eingabe verschreiben sollten, können Sie wie erwähnt frei editieren. Mit RETURN schließen Sie die Eingabe ab.

Wenn das FLAG gesetzt ist (hier muß (!) - nach dem Komma - eine numerische Variable stehen), kann die Eingabe außer mit RETURN auch mit CURSOR auf/ab abgeschlossen werden. Der FLAG-Variablen wird dann ein entsprechender Wert übergeben:

CURSOR auf	=	-1
RETURN	=	0
CURSOR ab	=	1

Wie können Sie nun diese zusätzlichen Option ausnutzen?

Stellen Sie sich vor, Sie wollen eine ganze Tabelle voller Zahlen eingeben. Sehen Sie sich dazu bitte das folgende kleine Programm an:

```
100 dim z$(10)
110 d$="0123456789"
120 z=5:s=15:1=10
130 :
140 sys 2096,z,s,1,d$,z$(z-5),f
150 if f=0 then500
160 if f=1 and z<15 then z=z+1:goto140
170 if f=-1and z>5  then z=z-1:goto140
180 goto140
190 :
500 rem weiteres Programm
```

Mit diesem Programm können Sie eine Zahlentabelle von 11 Zahlen eingeben und solange Veränderungen jeder Art vornehmen, bis Sie die RETURN-Taste drücken. (Zeile 150 prüft auf RETURN und verzweigt dann zum weiteren Programm.)

Mit etwas Phantasie können Sie mit diesem INKEY-Befehl sehr effiziente und vor allem sichere und anwenderfreundliche Programme "zaubern".

Zum Schluß noch eine Anregung zum Experimentieren: Der alte Zielstring könnte doch vielleicht für einen neuen Zielstring der Definitionsstring sein - oder?

Haushaltsbuchführung:

Wer hat sich nicht schon oft genug über seine ungeordneten Finanzen geärgert? Die Haushaltsbuchführung macht jetzt Schluß damit.

Dieses Programm verwaltet etwa 580 Haushaltsbelege nach Einnahmen und Ausgaben. Die Ausgabe der Belege erfolgt auf Wunsch entweder nur auf dem Bildschirm, oder auf Bildschirm und Drucker. Dieser kann ein Matrixdrucker (zum Beispiel MPS 801) oder auch ein Plotter (1520) sein. Die Wahl der Geräteadresse erfolgt softwaremäßig.

Als Speichermedien eignen sich sowohl Datasette als auch Diskette. Nachdem Sie das Programm aus INPUT 64 auf Ihren eigenen Datenträger geSAVEd und anschließend wieder geladen haben, können Sie das Programm mit RUN starten. Es erscheint das Hauptmenue.

Gleichzeitig werden angezeigt:
- der freie Speicherplatz
- die Anzahl der noch möglichen Einträge. Gegebenenfalls erscheint der Hinweis: (vor-)letzte Dateneingabe.

Durch Drücken der jeweiligen Funktionstaste gelangt man zu:

1. EINGABE DER BELEGE

Die Eingabeseite erscheint. Die aktuelle Belegnummer wird rechts oben dreistellig vorgegeben.

Es folgen von oben links die Eintragungen nach:
- Datum, - Betrag, - Bereich/Artikel, - Einnahme/Ausgabe.

Eine Korrektur fehlerhafter Eingaben ist durch die INST/DEL-Taste in jeder Zeile möglich. Am Ende jeder Zeile erfolgt ein Signalton und die Abfrage 'OK?'. Jetzt kann noch korrigiert oder durch Drücken der RETURN-Taste in die nächste Zeile gesprungen werden. Auch vor dem eigentlichen Ende einer Zeile kann weitergesprungen werden. Der Zeilenrest wird dann durch Punkte gefüllt.

Falls Sie in der Eingabezeile 'BETRAG' statt des Dezimalpunktes fälschlicherweise ein Komma setzen, wird dieses automatisch berichtigt. Bei der Eingabe von Beträgen über 9999.99 sind keine Pfennigeingaben mehr möglich.

In der Eingabezeile 'BER./Art.' sind die ersten fünf Stellen besonders gekennzeichnet. Hier ergibt sich die Möglichkeit, bei der Belegablage Bereichsfelder zu bilden. Anregungen für Belegbereiche im Fall 'HAUSHALT' sind auf der Eingabeseite unten dargestellt. Bei einer späteren Ausgabe können diese Bereiche aufgerufen werden.

Eine Einnahme wird durch '+', eine Ausgabe durch '-' eingegeben. Will man eine Eingabe kostenneutral einbringen, kann dies beispielsweise durch '0' oder ähnliches ausgedrückt werden.

Hilfen bei der Eingabe werden direkt auf der Bildschirmseite gegeben:

Bei Betätigen der Taste CLR/HOME wird die Eingabe verlassen und ins Menue zurückgekehrt. Dies ist unschädlich für den Programmablauf beziehungsweise die Vorgabe der Belegnummer.

Nach Eingabe aller Zeilen und dem letzten RETURN springt das Programm in die Prüfroutine. Hier wird überprüft, ob in der Betragszeile ein Punkt gesetzt wurde und ob in der EIN./AUSG. '+' oder '-' steht. Ist dies der Fall, wird die Eingabe als DATA-Zeile am Ende des Programms abgelegt. Im anderen Fall wird auf den fehlerhaften Sachverhalt hingewiesen, wobei die Möglichkeit der Neueingabe oder bei beabsichtigter 'Falscheingabe' des Speicherns besteht.

Auch hier ist das Springen ins Menue unschädlich für die eingegeben Daten.

Wird die Neueingabe gewählt, erscheint zwar wieder die Eingabeseite, aber so, daß nicht nur die Belegnummer wieder oben rechts erscheint, sondern auch der zuvor eingegebene Text erscheint revers im ersten Querbalken. Dies soll die Korrektur erleichtern.

2. AUSGABEN NACH KRITERIEN

Nach Drücken der Funktionstaste F3 erscheint die Frage nach dem Ausgabekriterium.

Hier kann gewählt werden zwischen:

- EINN./AUSG.
- MONAT/JAHR
- JAHR
- TAGESDATUM
- BETRAG
- BER./ARTIKEL
- BELEG-NR.

Wenn Sie das Ausgabekriterium 'BER./ARTIKEL' gewählt haben, ist folgendes zu beachten:

Es stehen wiederum drei Möglichkeiten zur Auswahl:

- L AUSWAHL V. LI. WAEHLBAR
- F AUSWAHL NACH 5 STELLEN
- Z AUSWAHL NACH 12 STELLEN

Nach der Auswahl erscheint, ähnlich der Eingabe, eine Punktzeile zur Beschriftung. Die Länge der Punktzeile richtet sich dabei nach dem Ausgabekriterium. Dieses erscheint nochmals zur Kontrolle am Kopfbalken. Zu beachten ist dabei folgendes:

Bei der Option 'L' sind nur die eingegebenen Zeichen relevant. Es werden alle BEREICHE/ARTIKEL aufgelistet, die mit diesen eingegebenen Zeichen anfangen.
Bei der Option 'F' werden nur die BEREICHE/ARTIKEL aufgelistet, die aus fünf Zeichen bestehen und mit der Eingabe identisch sind.

Bei der Option 'Z' werden alle BEREICHE/ARTIKEL aufgelistet, die mit den ersten fünf eingegebenen Zeichen übereinstimmen.

Durch Drücken der RETURN-Taste, wird die Eingabe abgeschlossen und die Frage nach der Ausgabeart eingeblendet.

Mit 'B' erfolgt die Ausgabe nur auf den Monitor, mit 'P' simultan auf Monitor und Drucker beziehungsweise Plotter.

Die Ausgabe auf Monitor oder Drucker kann jederzeit durch Drücken der RUN/STOP-Taste abgebrochen werden. In diesem Fall muß das Programm entweder neu gestartet oder mit CONT weitergeführt werden.

Fehlermeldungen (DEVICE NOT PRESENT oder ähnlich) erscheinen nicht auf dem Bildschirm.

Auch das Ausgabekriterium 'BELEG NR.' bietet drei Möglichkeiten:

- EINZELBELEG
- BELEGFOLGE
- ALLE BEFEHLE

Die Abfrage nach dem Einzelbeleg erfordert keine weiteren Hinweise.

Bei der Belegfolge müssen die Anfangsbelegnummer und die Endbelegnummer eingegeben werden. Die letzte Belegnummer wird angezeigt.

3. KORREKTUR

Korrigiert werden kann nur nach Belegnummern. Nach Anwahl dieser Funktion erscheint die Abfrage nach der Belegnummer. Haben Sie die Nummer eingegeben und durch die RETURN-Taste bestätigt, wird der genannte Beleg zur Kontrolle im reversen Querbalken eingeblendet.

Nun stehen die Neueingabe, oder bei einer irrtümlichen Eingabe, ein Weitersuchen oder die Rückkehr ins MENUE zur Auswahl.

Bei 'NEUEINGABE' erscheint wieder die Eingabeseite. Zur Kontrolle beziehungsweise zum Vergleich ist im ersten Querbalken die alte Belegung aufgeführt. Die Belegnummer erscheint wieder oben rechts.
Alles weitere läuft wie bei 'EINGABE'.

4. ABSPEICHERN/VERIFIZIEREN

Das Abspeichern kann sowohl auf Kassette, als auch auf Diskette erfolgen. Zur Auswahl erscheinen nach Anwahl der Speicherfunktion die entsprechenden Hinweise.

Nach erfolgtem 'SAVEN' kann auch noch verifiziert werden. Die Aufforderung dazu erscheint selbsttätig.

Noch eine Bemerkung zu 'EINGABE':

Bei der Eingabe des Datums wird nicht abgefragt, ob sinnvolle Daten für Tag und Monat eingegeben wurden. Bei solchen Eingaben stürzt der Rechner zwar nicht ab, aber es sieht bei der Ausgabe wenig sinnvoll aus, wenn auf dem Monitor oder Drucker das Datum: 84.67.85 erscheint.

Mathe mit Nico

Sie können Ihre Kenntnisse im Bereich der Geometrie auffrischen. Nico wird auch diesmal ein geduldiger Trainer sein.

Nico wird mit Ihnen Umfangs- und Flächenberechnungen von Rechtecken, Dreiecken und Kreisen üben.

Da wir Ihnen nicht zumuten wollten, Ergebnisse nebenher mit einem Taschenrechner oder schriftlich auf Papier nebenher zu berechnen, können Sie jederzeit mit "R" ein Rechenblatt aufrufen.

Die folgenden Rechenoperationen stehen Ihnen dann zur Verfügung:

+	= Plus	Beispiel:	25 + 47
-	= Minus		61 - 25.1
*	= Mal		52 * 33.9
/	= Geteilt durch		12 / 3.5
↑	= Quadrat		4.5↑
←	= Quadratwurzel		25 ←

Die Rechenoperation müssen Sie mit einem RETURN-Zeichen abschließen. Sie können auch mehrere Berechnungen in einer Zeile ausführen lassen (Klammern sind aber leider nicht möglich).

Wenn Sie ein "L" eingeben, können Sie Ihr Ergebnis auf die Aufgabenseite übertragen.

WETTBEWERBS-SIEGER: RECORDER-JUSTAGE

ZUM AUTOR

Harald Diebek (29) beschäftigt sich seit über 10 Jahren mit Computern. Angefangen hat er auf einem KIM1, den er heute noch besitzt. Nach dem Studium der Elektrotechnik und der Physik ist er heute in Duisburg als Computer-Fachberater tätig. Seine Spezialität ist Programmierung in 6502-Assembler. Grundlagen erarbeitete er sich schon als Kind: Zum elften Geburtstag bekam er einen Elektronik-Baukasten geschenkt.

Die 3000 Mark gingen diesmal an den Autor eines nicht einmal zwei KByte langen Programms. Aber es geht ja auch nicht um Quantität, sondern um Qualität:

1.) RECORDER-JUSTAGE ist eine Programmneuheit. Bisher existiert unseres Wissens für den C 64 nichts Vergleichbares, obwohl alle Datasetten-Besitzer ein Lied von den Leseschwierigkeiten wegen verstellter Tonköpfe singen können. Unser eigenes JUSTAGE-Programm (das Listing hinten im Beiheft) wird, das geben wir ganz offen zu, von RECORDER-JUSTAGE um Längen übertroffen.

2.) Mit der RECORDER-JUSTAGE können Sie jede (!) Kassette mit Ihrem Bandgerät laden. Der Tonkopf kann mit Hilfe dieses Programms exakt auf das zu lesende File justiert werden, ganz gleich, mit welchem Aufnahmeverfahren aufgezeichnet wurde oder ob die Kassette normgerecht oder auf einer total verstellten Datasette aufgenommen wurde. Der Zeitaufwand für das Einstellen beträgt, abgesehen von der Ladezeit der RECORDER-JUSTAGE, weniger als eine halbe Minute.

3.) Die RECORDER-JUSTAGE ist - ohne Übertreibung - excellent programmiert. So erzeugt das sehr kurze Initialisierungsprogramm ein weiteres Programm mit einer Länge von über 4 KByte. Das bei der vom Programm zu lösenden Aufgabe äußerst wichtige Timing wird exakt eingehalten. Da fällt auch eine kleine programmtechnische Unsauberkeit wie selbstverändernder Code in der Hauptroutine nicht ins Gewicht.

BEDIENUNG

Sie können die RECORDER-JUSTAGE sowohl innerhalb des Magazins anwenden als auch das Programm auf eigenen Datenträger abspeichern und separat laden und nutzen.

Innerhalb von INPUT 64 erscheint zunächst eine Textseite mit Erläuterungen. Drücken Sie die SHIFT-Taste, wird die eigentliche RECORDER-JUSTAGE gestartet. (Ist dieser Programmteil aktiviert, kann das INPUT 64-Betriebssystem nicht aufgerufen werden - kein CTRL-h, kein CTRL-i und so weiter !) Zum Ausstieg siehe unten.

Der Bandmotor läuft an, und Sie sehen folgende Hinweise auf der oberen Bildschirmhälfte:

1. Band einlegen und starten

2. Schraube bis zum Anschlag eindrehen

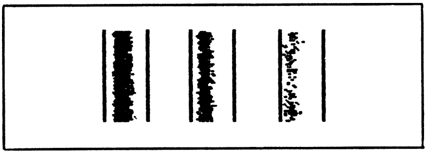
3. Schraube lösen, bis im Anzeigenfeld
   saubere Linien sichtbar werden

4. f1 - Hilfslinien für Commodore-Format
   f3 - Hilfslinien für SuperTape-Format

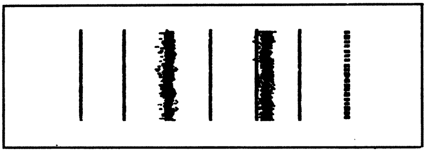
5. Mit RETURN Programm verlassen

Mit "Anzeigefeld" ist die untere Bildschirmhälfte gemeint. Dort sehen Sie zunächst nichts. Die genannte Schraube ist die Einstellschraube Ihres Daten-Recorders. Sie ist an der Commodore-Datasette durch die kleine Bohrung vor der REWIND-Taste erreichbar, allerdings nur bei gedrückter PLAY-Taste!

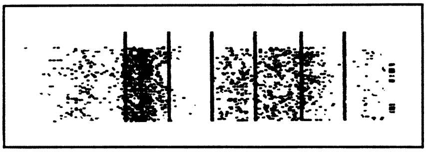
Legen Sie nun ein Band ein - es kann auch die INPUT-Kassette sein - und starten den Recorder (PLAY-Taste). Auf der unteren Bildschirmhälfte baut sich nun eine Grafik auf. Diese kann, je nach Aufzeichnungsverfahren und Einstellung des Tonkopfes, folgendes Aussehen haben:



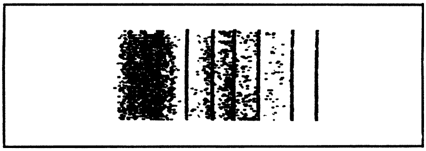
Richtige Einstellung: Die drei Messlinien sind schmal und innerhalb der Hilfslinien



Richtig eingestellter Tonkopf, aber falsche Bildschirmaufzeichnung: Cursor-left oder f1 Drücken!



Falsch: Unsaubere Messlinien, weit außerhalb der Hilfslinien.



Total verstellt und falsches Aufzeichnungsformat gewählt: Zu breite Linien, außerdem im falschen Bereich!

Die Linien müssen möglichst scharf und sauber erscheinen. Ist dies nicht der Fall, drehen Sie zunächst die Einstellschraube vorsichtig bis zum Anschlag ein (im Uhrzeigersinn). Dann muß die Schraube Stück für Stück gelöst werden, bis die optimale Einstellung erreicht ist. Üben Sie dabei möglichst keinen Druck auf den Schraubenkopf aus, weil die Mechanik sehr nachgiebig ist.

Zur Erleichterung können Sie Hilfslinien anwählen; mit f1 die für Commodore-Files, mit f3 für SuperTape.

Mit der RECORDER-JUSTAGE können Sie aber auch jedes beliebige andere Aufzeichnungsformat auswerten. Zur Unterscheidung zwischen "schnellen" und "langsamen" Ladeverfahren dienen die Cursortasten: Cursor-down schiebt die Linien nach links für langsame Verfahren, Cursor-right für schnelle (außer SuperTape eventuell noch Turbo-Tape oder ähnliches.)

Das Programm verlassen können Sie durch Betätigen der RETURN-Taste. Innerhalb des Magazins kommen Sie dadurch zurück auf die erläuternden Textseiten. Hatten Sie die RECORDER-JUSTAGE von Ihrem eigenen Datenträger geladen, bewirkt dieser Ausstieg einen BASIC-Warmstart - vergleichbar dem Effekt beim gleichzeitigen Betätigen von RUN/STOP und RESTORE.

ABSPEICHERN AUF EIGENEN DATENTRÄGER

Mit CTRL und s kann RECORDER-JUSTAGE wie gewohnt aus dem Magazin heraus abgespeichert werden. Dies ist nur während der Erläuterungsseiten möglich, da im eigentlichen Programm das INPUT 64-Betriebssytem "abgehängt" ist.

FÜR TECHNISCH INTERSSIERTE

BEGRIFFE

INTERRUPT nennt man eine gesteuerte Unterbrechung des gerade abgearbeiteten Programms und den Sprung in eine andere Routine. Im C 64 wird serienmäßig dieser Interrupt alle 1/20 Sekunde durchgeführt (Timer-gesteuert also, nach Ablauf einer definierten Zeit), um die Tastatur abzufragen, TI zu erhöhen und so weiter.

FLAG-INTERRUPT wird unter anderem genutzt zum Lesen vom Band. Die zuständige Adresse ist Register 13 des ersten Ein-Ausgabe-Bausteins (CIA 1, Basisadresse $DC00). Wird dort Bit 4 und gleichzeitig Bit 7 gesetzt, werden negative Flanken vom Kassettenport zur Interrupt-Quelle. Löscht man Bit 7 dieses Register, werden die Flag-Interrupts verhindert.

RASTERZEILE ist die Zeilenposition des Elektronenstrahls, der das Monitor- oder Fernseherbild aufbaut. Der obere Rand des Schirms entspricht dabei etwa Position 32, der untere Position 260.

RASTERZEILEN-INTERRUPT kann durch Beschreiben bestimmter Register des Video-Chips ermöglicht werden. Dadurch löst der VIC an festlegbaren Positionen des Elektronenstrahls einen Interrupt aus. (siehe auch HILFS-PROGRAMME 2 in dieser Ausgabe und BITS & BYTES IM VIDEO-CHIP Teil 5 in der letzten Ausgabe)

DAS PRINZIP

Die Arbeitsweise von RECORDER-JUSTAGE beruht auf der Auswertung der Laufzeiten zwischen Flankenwechseln des Signals am Kassettenport.

Die technische Voraussetzung dafür liegt in der "Interruptfähigkeit" des I/O-Chips 6526 (auch CIA = Central Interface Adapter genannt). Dieser kann so programmiert werden, daß negative Flanken am Kassettenport einen Interrupt auslösen.

Ebenso kann auch der Video-Chip durch entsprechende Programmierung Interrupts auslösen, und zwar bei einer definierbaren Zeilenposition des Elektronenstrahls. Des weiteren muß man wissen, daß der Video-Chip während des Bildaufbaus zeitliche Priorität vor dem Prozessor hat, diesen also gegebenenfalls "anhalten" kann. Deswegen wird bei Kassettenoperationen das Bild ausgeschaltet, damit ein ungestörtes Timing bei der Datenübertragung gewährleistet ist.

Die Besonderheit von RECORDER-JUSTAGE besteht darin, daß der Autor die verschiedenen Interrupt-Quellen kombiniert. Bestimmend ist der Rasterzeilen-Interrupt. Er wird zum einen genutzt, um in der oberen Bildschirmhälfte normalen Text und in der unteren Hälfte hochauflösende Grafik darzustellen.

Entscheidender ist, daß über die Strahlposition erkannt werden kann, ob der Elektronenstrahl gerade das Bild aufbaut oder zurückläuft. In dieser Dunkelphase wird natürlich der Prozessor nicht vom VIC unterbrochen, und eine exakte Auswertung von Signalen am Kassettenport ist möglich. So werden Interrupts von der CIA 2 nur in der Dunkelphase zugelassen, nur jetzt finden die Zeitvergleiche statt. Während des Bildaufbaus werden Interrupts durch Port-Signale verhindert, da in dieser Zeit wegen möglicher Störungen durch den VIC die Meßgenauigkeit beeinträchtigt wäre.

Der Rest ist "nur" noch eine geschickte graphische Umsetzung der Auswertung der Signalabstände vom Kassettenport. Je länger die verstrichene Zeit, das heißt, je größer die Leseungenauigkeit wegen nicht erkannter Flanken, desto breiter ist die Streuung der Linien auf dem Schirm.

IM DETAIL

In der Initialisierungsroutine (Label ENTRY) werden vom Programm benutzte Zero-Page-Vektoren gerettet bzw. neu gesetzt sowie der Interruptvektor auf eine eigene Routine (Einsprung: Label IRQENT) "verbogen".

Flag-Interrupts werden verhindert, nur noch der Video-Chip löst gezielte Programmunterbrechungen bei einer festgelegten Zeilenposition des Elektronenstrahls aus. (Zeile 2280 bis 2300 und 3440 bis 3470) Die Timer werden mit einem festen Wert geladen, um später exakte Zeitvergleiche durchführen zu können.

Außerdem wird die Scroll-Routine für den Hires-Bildschirm erzeugt (SCRGEN) und eine Warteschleife (SCHLGEN), die Grafik initialisiert und so weiter.

Erreicht nun der Elektronenstrahl beim Aufbau des Bildschirms die Position 32 ($20), wird ein Interrupt ausgelöst, der nach IRQENT springt. Nach der Entscheidung, ob es ein Rasterzeilen- oder ein Timer-Interrupt war (Zeile 4380 bis 4390), geht es, da FLAG noch Null ist, nach RAS1. Die Tastatur wird abgefragt und die Scroll-Routine begonnen. (Zeile 3730 ff.)

Der Video-Chip unterbricht wieder, wenn der Elektronenstrahl die Position 177 ($B1) erreicht. Dies wurde zu Beginn von RAS1 festgelegt. (Zeile 3560 bis 3580) Es wird auf hochauflösende Grafik für die Darstellung der Messlinien in der unteren Bildschirmhälfte umgeschaltet. Über RTI geht es zurück zum Scrollen.

Die dritte Rasterzeilen-Interrupt-Position RAS3 ist der Beginn der eigentlichen Messung. Der sogenannte FLAG-Interrupt wird wieder zugelassen (Zeile 4220 f.). Das heißt, jedes empfangene Signal vom Kassettenport löst einen Interrupt aus, der in eine Messschleife (Zeile 4510 ff.) führt. Dort wird die seit dem letzten Flag-Interrupt - sprich dem letzten von Kassette empfangenen Signal - verstrichene Zeit ausgewertet und auf dem Grafik-Bildschirm entsprechende Punkte gesetzt.

Als neue interrupt-auslösende Rasterzeilen-Position wurde zu Beginn von RAS3 der Wert 32 ($20) in das entsprechende Video-Chip-Register geschrieben (Zeile 4150 bis 4160). Beim nächsten Interrupt des VIC's geht es von vorn los bei RAS1. Die FLAG-Interrupts werden zwischen RAS1 und RAS3 nicht zugelassen, da in dieser Zeit der Video-Chip das Monitor-Bild aufbaut und dabei zeitweilig den Prozessor stoppt. Dies würde zu Messfehlern führen.

Weiter Interessierten sei das Studium des nachfolgenden Assembler-Listings empfohlen.

SPEICHERBELEGUNG

(nach Start durch RUN)

$8000	$85F6	Hauptroutine
$6000	$7FFF	Video-RAM Hires-Screen
$5000	$5FFF	Scroll-Routine, wird generiert
$4910	$4911	Back-Up für IRQ-Vektor
$4900	$490F	Back-Up für Zero-Page-Adressen
$4800	$48FF	Warteschleife, wird generiert
$4400	$47FF	Farbspeicher zu Hires-Screen

2

```
profi-ass 64 v2.0     seite 1

910:   8000               .opt p4,o1
1000:               ;
1010:               ; zero page adressen
1020:               ;
1030:  00c0         lstkey    =    $c0
1040:  00c1         pntrl     =    $c1
1050:  00c2         pntlh     =    $c2
1060:  00c3         c3        =    $c3
1070:  00c4         c4        =    $c4
1080:  00c5         opc1      =    $c5
1090:  00c6         opc2      =    $c6
1100:  00c7         s3        =    $c7
1110:  00c8         s4        =    $c8
1120:  00c9         s5        =    $c9
1130:  00ca         s6        =    $ca
1140:  00cb         mzael     =    $cb
1150:  00cc         scrflg    =    $cc
1160:  00cd         stkptr    =    $cd
1170:  00ce         flag      =    $ce
1180:               ;
1190:               ; i/o adressen
1200:               ;
1210:  d011         cnt       =    $d011    ; vic modus
1220:  d012         zei       =    $d012    ; zeilen# for irq
1230:  d018         ads       =    $d018    ; adressen
1240:  d019         ifr       =    $d019    ; vic interrupt flags
1250:  d01a         imr       =    $d01a    ; vic interrupt mask
1260:  d020         raf       =    $d020    ; randfarbe
1270:  d021         blf       =    $d021    ; blattfarbe
1280:  dc00         kbr       =    $dc00    ; tastatur zeile
1290:  dc01         kbc       =    $dc01    ; tastatur spalte
1300:  dc04         tll       =    $dc04    ; timer low order
1310:  dc05         tlh       =    $dc05    ; timer hi order
1320:  dc0d         icr       =    $dc0d    ; interrupt controll
1330:  dc0e         cr1       =    $dc0e    ; timer controll
1340:  dd00         blo       =    $dd00    ; vic 16k block
1350:  dd0d         ics       =    $dd0d    ; interrupt controll
1360:               ;
1370:               ; externe labels
1380:               ;
1385:  able         print     =    $able
1390:  4801         schleife  =    $4801
1400:  5000         scroll1   =    $5000
1410:  5800         scroll2   =    $5800
1420:  ff81         vidres    =    $ff81
1430:  ff84         initcia   =    $ff84
1440:  ff8a         initio    =    $ff8a
1450:  4900         platz     =    $4900
1460:  4910         platz2    =    platz+16
1465:  2000         menue     =    $2000
1470:               ;
1480:               ; programm einsprung
1490:               ;
1500:  8000                   *=   $8000     ; einsprung-vektor
1510:  8000 4c 4c 80          jmp  entry
1520:               ;
1530:               ; vektoren
1540:               ;
```

```
profi-ass 64 v2.0     seite 2

1560:  8003 72 81    nvek     .wordirqent  ;irq vektor (raster,flag)
1570:                ;
1580:                ; konstante
1590:                ;
1610:  8005 c0 60 30 kontbl   .byte$c0,$60,$30,$18
1620:  8009 0c 06 03          .byte$0c,$06,$03,$01
1630:                ;
1650:  800d bd       sour1    .byte$bd      ; lda abs,x
1660:  800e 9d                .byte$9d      ; sta abs,x
1670:  800f f8 7d             .word$7df8
1680:  8011 f9 7d             .word$7df9
1690:                ;
1710:  8013 ad       sour2    .byte$ad      ; lda abs
1720:  8014 8d                .byte$8d      ; sta abs
1730:  8015 bf 7c             .word$7cbf
1740:  8017 f8 7d             .word$7df8
1750:                ;
1770:  8019 ea       du2      nop
1780:  801a ea                nop
1790:  801b ea                nop
1800:                ;
1820:  801c 4a       du4      lsr  a
1830:  801d 66 c3             ror  c3
1840:                ;
1850:                ; einsprung restore-vektor
1860:                ;
1870:                ; alles zuruecksetzen
1880:                ; und programm verlassen
1890:                ;
1910:  801f 48       nmient   pha
1920:  8020 8a                txa
1930:  8021 48                pha
1940:  8022 98                tya
1950:  8023 48                pha
1960:  8024 ba                tsx
1970:  8025 bd 05 01          lda  $0105,x
1980:  8028 18                clc
1990:  8029 69 02             adc  #$02
2000:  802b 9d 05 01          sta  $0105,x
2010:  802e 90 03             bcc  break
2020:  8030 fe 06 01          inc  $0106,x
2030:                ;
2040:                ; restauriere c0 bis cf und vektoren
2050:                ;
2070:  8033 20 07 82 break    jsr  exit
2080:                ;
2090:  8036 20 84 ff          jsr  initcia  ; cia's initialisieren
2100:                ;
2110:  8039 20 8a ff          jsr  initio   ; i/o initialisieren
2120:                ;
2130:  803c 20 81 ff          jsr  vidres   ; video reset
2140:                ;
2142:  803f ac 83 c4          ldy  50307    ; input 64 !!
2143:  8042 c0 78             cpy  #$78     ; ist sei
2144:  8044 f0 03             beq  inpout
2146:                ;
2150:  8046 6c 02 a0          jmp  ($a002)  ; verlasse programm
2152:                ;
```

```
profi-ass 64 v2.0    seite 3

2155:  8049 4c 00 20 inpout   jmp  menue
2160:                ;
2170:                ; programm-einsprung
2180:                ;
2190:  804c 78       entry    sei            ; disable interrupts.
2195:  804d 20 80 84          jsr  txtout    ; zum hinweise schreiben
2200:  8050 d8                cld
2210:  8051 ba                tsx            ; merke
2220:  8052 86 cd             stx  stkptr    ; stack pointer.
2230:                ;
2240:  8054 20 eb 81          jsr  init      ; rette c1-cf
2250:                ;            rette vektoren
2260:                ;            setze neue vektoren
2270:                ;
2280:  8057 a9 7f             lda  #$7f      ; disable 6522
2290:  8059 8d 0d dc          sta  icr       ; irq and
2300:  805c 8d 0d dd          sta  ics       ; nmi
2310:                ;
2320:  805f a2 00             ldx  #$00      ; latch timer 1
2330:  8061 a0 80             ldy  #$80      ; auf 8000
2340:  8063 8e 04 dc          stx  t1l
2350:  8066 8c 05 dc          sty  t1h
2360:                ;
2370:  8069 a5 01             lda  $01       ; motor
2380:  806b 29 df             and  #%11011111
2390:  806d 85 01             sta  $01       ; on
2400:                ;
2410:  806f a2 0a             ldx  #$0a      ; setze
2420:  8071 8e 20 d0          stx  raf       ; randfarbe
2430:  8074 e8                inx            ; und
2440:  8075 8e 21 d0          stx  blf       ; blattfarbe
2450:                ;
2460:  8078 20 74 82          jsr  scrgen    ; generiere scroll-routine
2470:  807b 20 1d 82          jsr  schlgen   ;generiere warteschleife
2480:                ;
2490:                ; farbmartrix 4400-47e7 besetzen
2500:                ;
2510:  807e a9 47             lda  #$47      ; oberste
2520:  8080 85 c2             sta  pnt1h     ; seite.
2530:  8082 a2 43             ldx  #$43      ; unterste seite-1.
2540:  8084 a9 0a             lda  #$0a      ; farben
2550:  8086 20 38 82          jsr  fillup
2560:                ;
2570:  8089 a2 7f             ldx  #$7f      ; cursor down taste
2580:                ;
2600:  808b 86 c0    marken   stx  lstkey
2610:                ;
2620:                ; clear videomatrix 6000-7f3f
2630:                ;
2640:  808d a9 7f             lda  #$7f      ; oberste
2650:  808f 85 c2             sta  pnt1h     ; seite.
2660:  8091 a2 5f             ldx  #$5f      ; unteste seite-1.
2670:  8093 a9 00             lda  #$00      ; inhalt fuer matrix
2680:  8095 20 38 82          jsr  fillup
2690:                ;
2700:  8098 a5 c0             lda  lstkey    ; wandele
2710:  809a a2 06             ldx  #$06      ; zahl
2720:  809c ca       sft      dex            ; in
```

```
profi-ass 64 v2.0    seite 5

3330:  80e1 bd 1c 80 dur4     lda  du4,x
3340:  80e4 9d a7 81          sta  div,x
3350:  80e7 ca                dex
3360:  80e8 10 f7             bpl  dur4
3380:                ;
3390:                ;
3400:                ; set up for raster irq
3410:                ;
3420:  80ea e8       to       inx            ; flag = $00
3430:  80eb 86 ce             stx  flag
3440:  80ed a9 20             lda  #$20      ; zei  = $20
3450:  80ef 8d 12 d0          sta  zei
3460:  80f2 e8                inx            ; imr  = $01
3470:  80f3 8e 1a d0          stx  imr
3480:  80f6 4c 01 48          jmp  schleife
3490:                ;
3500:  80f9 d0 90    markenv  bne  marken
3510:                ;
3520:                ; einsprung aus raster 1
3530:                ; und aus stack ueberlauf
3540:                ;
3560:  80fb e6 ce    ras1     inc  flag
3570:  80fd a9 b1             lda  #$b1
3580:  80ff 8d 12 d0          sta  zei
3590:                ;
3600:                ; disable flag irq
3610:                ;
3620:  8102 a9 10             lda  #$10
3630:  8104 8d 0d dc          sta  icr
3640:  8107 ad 0d dc          lda  icr       ;lese blind
3650:                ;
3660:                ; stack neu initialisieren
3670:                ;
3680:  810a a6 cd             ldx  stkptr
3690:  810c 9a                txs
3700:                ;
3710:                ; tastatur abfragen
3720:                ;
3730:  810d a9 fe             lda  #$fe      ; aktiviere tastatur
3740:  810f 8d 00 dc          sta  kbr       ; zeile 1
3750:  8112 ae 01 dc          ldx  kbc
3760:  8115 e0 fe             cpx  #$fe      ; sprung wenn keine
3770:  8117 b0 0e             bcs  nokey     ; taste oder 'del'
3780:  8119 e4 c0             cpx  lstkey    ; ignoriere falls noch
3790:  811b f0 0a             beq  nokey     ; immer selbe taste
3800:  811d e0 fd             cpx  #$fd
3810:  811f d0 d8             bne  markenv   ; echte taste
3820:  8121 a6 cd             ldx  stkptr
3830:  8123 9a                txs
3840:  8124 4c 33 80          jmp  break
3850:                ;
3860:                ;
3870:  8127 20 61 82 nokey    jsr  hiresof
3880:                ;
3890:                ; scrollen
3900:                ;
3910:  812a 58                cli
3920:  812b a5 cc             lda  scrflg
```

```
2730:  809d 2a              rol  a         ; bitmuster
2740:  809e b0 fc           bcs  sft       ; um
2750:                  ;
2760:                  ; cursor unten  --- x=5
2770:                  ; cursor rechts --- x=0
2780:                  ; f1--x=2       f3--x=3
2790:                  ; f5--x=4       f7--x=1
2800:                  ;
2810:  80a0 8a              txa            ; x=x*2
2820:  80a1 0a              asl  a
2830:  80a2 aa              tax
2840:                  ;
2850:                  ; set up for markieren
2860:                  ;
2870:  80a3 bd 16 83        lda  anftab,x   ; pntr1 auf erstes
2880:  80a6 85 c1           sta  pntr1      ; byte der jeweiligen
2890:  80a8 e8              inx            ; marken tabelle
2900:  80a9 bd 16 83        lda  anftab,x
2910:  80ac 85 c2           sta  pnt1h
2920:                  ;
2930:                  ; anzahl der tabellenbytes holen
2940:                  ;
2950:  80ae a0 00           ldy  #$00
2960:  80b0 b1 c1           lda  (pntr1),y
2970:  80b2 85 ce           sta  flag       ;
2980:  80b4 0a              asl  a
2990:  80b5 a8              tay
3000:                  ;
3010:  80b6 f0 16           beq  nom       ; sprung, wenn keine marken mehr
3020:  80b8 b1 c1      mark lda  (pntr1),y
3030:  80ba 0a              asl  a
3040:  80bb 0a              asl  a         ; das bitmuster
3050:  80bc 0a              asl  a         ; landet
3060:  80bd aa              tax            ; letztlich
3070:  80be a9 00           lda  #$00      ; in
3080:  80c0 69 72           adc  #$72      ; $72c7+8*spalte
3090:  80c2 8d ca 80        sta  desthi
3100:  80c5 88              dey
3110:  80c6 b1 c1           lda  (pntr1),y  ; hole bitmuster
3120:                  ;
3130:  80c8 9d              .byte$9d       ; setze bitmuster
3140:  80c9 c7              .byte$c7       ; sta abs,x
3150:  80ca 72      desthi  .byte$72
3160:                  ;
3170:  80cb 88              dey
3180:  80cc d0 ea           bne  mark
3190:                  ;
3200:                  ; setze division
3210:  80ce 26 ce   nom     rol  flag       ; des messwertes
3220:  80d0 90 0d           bcc  durch4     ; durch 2 oder
3230:  80d2 a2 02           ldx  #$02       ; durch 4, je nach
3240:  80d4                                ; flag
3250:  80d4 bd 19 80 dur2   lda  du2,x
3260:  80d7 9d a7 81        sta  div,x
3270:  80da ca              dex
3280:  80db 10 f7           bpl  dur2
3290:  80dd 30 0b           bmi  to
3310:  80df a2 02   durch4  ldx  #$02
```



```
3930:  812d 49 ff              eor  #$ff
3940:  812f 85 cc              sta  scrflg
3950:  8131 30 06              bmi  scr1
3960:  8133 20 00 50           jsr  scroll1
3970:  8136 4c 36 81           jmp  *
3980:  8139 20 00 58 scr1      jsr  scroll2
3990:  813c 4c 3c 81           jmp  *
4000:                  ;
4010:                  ; einsprung raster 2
4020:                  ;
4040:  813f e6 ce     ras2     inc  flag
4050:  8141 a9 ff              lda  #$ff
4060:  8143 8d 12 d0           sta  zei
4070:  8146 20 48 82           jsr  hireson
4080:  8149 4c e5 81           jmp  rfi
4090:                  ;
4100:                  ; einsprung raster 3
4110:                  ;
4130:  814c a9 00     ras3     lda  #$00
4140:  814e 85 ce              sta  flag
4150:  8150 a9 20              lda  #$20
4160:  8152 8d 12 d0           sta  zei
4170:                  ;
4180:  8155 a6 cd              ldx  stkptr
4190:  8157 9a                 txs
4200:                  ;
4210:  8158 e6 cb              inc  mzael    ; ignoriere 1. messung
4220:  815a a9 90              lda  #$90     ; enable
4230:  815c 8d 0d dc           sta  icr      ; flag irq
4240:  815f ad 0d dc           lda  icr
4250:  8162 4c 01 48           jmp  schleife
4260:                  ;
4270:                  ; verteiler raster irq's
4280:                  ;
4290:  8165 8d 19 d0 raster    sta  ifr
4300:  8168 a5 ce              lda  flag     ; austastluecke
4310:  816a f0 8f     raslv    beq  ras1
4320:  816c c9 01              cmp  #$01
4330:  816e f0 cf              beq  ras2
4340:  8170 d0 da              bne  ras3
4350:                  ;
4360:                  ; irq haupteinsprung
4370:                  ;
4380:  8172 ad 19 d0 irqent    lda  ifr
4390:  8175 30 ee              bmi  raster
4400:                  ;
4410:                  ; hier flag irq
4420:                  ; stop timer 1
4430:                  ;
4440:  8177 ad 0d dc           lda  icr
4450:  817a a9 88              lda  #$88
4460:  817c 8d 0e dc           sta  cr1
4470:                  ;
4480:                  ; 19 mik.sec seit flag irq
4490:                  ; berechne zeit
4500:                  ;
4510:  817f 38                 sec
4520:  8180 a9 19              lda  #$19
```

```
profi-ass 64 v2.0    seite 7

4530:  8182 ed 04 dc          sbc  t1l
4540:  8185 85 c3             sta  c3
4550:  8187 a9 80             lda  #$80
4560:  8189 ed 05 dc          sbc  t1h
4570:                ;
4580:  818c a2 99             ldx  #$99     ; timer 1
4590:  818e 8e 0e dc          stx  cr1      ; neustart
4600:                ;
4610:                ; ignoriere erste messung
4620:                ;
4630:  8191 a4 cb             ldy  mzael
4640:  8193 f0 05             beq  goon
4650:  8195 c6 cb             dec  mzael
4660:  8197 4c e5 81          jmp  rfi
4670:                ;
4690:  819a ba       goon      tsx
4700:  819b e0 10             cpx  #$10
4710:  819d b0 07             bcs  di
4720:  819f a9 ff             lda  #$ff
4730:  81a1 8d 00 74          sta  $7400
4740:  81a4 d0 c4             bne  raslv
4750:                ;
4760:  81a6 58       di        cli
4770:  81a7 4a       div       lsr  a        ; division
4780:  81a8 66 c3             ror  c3         ; durch 4
4790:  81aa 4a                lsr  a          ; oder  2
4800:  81ab 66 c3             ror  c3         ;
4810:  81ad a6 c3             ldx  c3         ; x=lo   a=hi
4820:                ;
4830:  81af c9 01             cmp  #$01     ;   test ob     hi < 1
4840:  81b1 90 0d             bcc  timok    ;   hi=0
4850:                ;
4860:                ; hier hi=1,2,3..
4870:                ;
4880:  81b3 d0 04             bne  over       ; nach over da hi>1
4890:                ;
4900:                ; hier hi=1
4910:                ;
4920:  81b5 e0 38             cpx  #$38     ;   hier hi=1
4930:  81b7 90 07             bcc  timok    ;   lo<40
4940:                ;
4950:                ; overflow
4960:                ;
4970:  81b9 a9 0f    over      lda  #$0f
4980:  81bb 8d 38 75          sta  $7538
4990:  81be d0 25             bne  rfi
5000:                ;
5010:                ; time ok
5020:                ;
5030:  81c0 69 74    timok     adc  #$74
5040:  81c2 85 c4             sta  c4
5050:  81c4 8a                txa
5060:  81c5 29 f8             and  #$f8
5070:  81c7 85 c3             sta  c3
5080:  81c9 8a                txa
5090:  81ca 29 07             and  #$07
5100:  81cc aa                tax
5110:  81cd bd 05 80          lda  kontbl,x
```

```
profi-ass 64 v2.0    seite 9

5710:  8225 20 38 82          jsr  fillup
5720:  8228 a9 58             lda  #$58     ; code for cli
5730:  822a 8d 01 48          sta  schleife
5740:  822d a9 d0             lda  #$d0     ; code for bne
5750:  822f 8d fe 48          sta  schleife+$fd
5760:  8232 a9 80             lda  #$80     ; code for sprungweite
5770:  8234 8d ff 48          sta  schleife+$fe
5780:  8237 60                rts
5790:                ;
5800:                ; fuelle seiten
5810:                ;
5830:  8238 a0 00    fillup    ldy  #$00
5840:  823a 84 c1             sty  pntr1
5860:  823c 88       fillu1    dey
5870:  823d 91 c1             sta  (pntr1),y
5880:  823f d0 fb             bne  fillu1
5890:  8241 c6 c2             dec  pnt1h
5900:  8243 e4 c2             cpx  pnt1h
5910:  8245 d0 f5             bne  fillu1
5920:  8247 60                rts
5930:                ;
5940:                ; hires screen on
5950:                ;
5970:  8248 a2 05    hireson   ldx  #$05
5980:  824a ca       again     dex
5990:  824b d0 fd             bne  again
6000:  824d ea                nop
6010:  824e a2 18             ldx  #$18
6020:  8250 a0 3b             ldy  #$3b
6030:  8252 ad 00 dd          lda  blo      ; waehle 2.
6040:  8255 29 fe             and  #$fe     ; 16k block
6050:  8257 8e 18 d0          stx  ads
6060:  825a 8c 11 d0          sty  cnt
6070:  825d 8d 00 dd          sta  blo      ; fuer vic
6080:  8260 60                rts
6090:                ;
6100:                ; hires screen off
6110:                ;
6130:  8261 a9 17    hiresof   lda  #$17
6140:  8263 8d 18 d0          sta  ads
6150:  8266 a9 1b             lda  #$1b
6160:  8268 8d 11 d0          sta  cnt
6170:  826b ad 00 dd          lda  blo      ; waehle 1.
6180:  826e 09 01             ora  #$01     ; 16k block
6190:  8270 8d 00 dd          sta  blo      ; fuer vic
6200:  8273 60                rts
6210:                ;
6220:  8274
7010:                ; scrollroutine generieren
6230:                ;
6250:  8274 a2 05    scrgen    ldx  #$05
6260:  8276 bd 0d 80 aga       lda  sour1,x
6270:  8279 95 c5             sta  opc1,x
6280:  827b ca                dex
6290:  827c 10 f8             bpl  aga
6300:                ;
6310:  827e a2 50             ldx  #$50
6330:  8280 86 c2    srg06     stx  pnt1h
```

```
5120:  81d0 11 c3     set     ora  (c3),y
5130:  81d2 91 c3             sta  (c3),y
5140:  81d4 e0 07             cpx  #$07
5150:  81d6 d0 0d             bne  rfi
5160:  81d8 e8                inx
5170:  81d9 98                tya
5180:  81da 69 08             adc  #$08
5190:  81dc a8                tay
5200:  81dd 90 02             bcc  skp
5210:  81df e6 c4             inc  c4
5220:  81e1 a9 80     skp     lda  #$80
5230:  81e3 d0 eb             bne  set
5240:  81e5 68        rfi     pla
5250:  81e6 a8                tay
5260:  81e7 68                pla
5270:  81e8 aa                tax
5280:  81e9 68                pla
5290:  81ea 40                rti
5300:                 ;
5310:                 ; backup $c0 bis $cf,
5320:                 ; backup vektoren und
5330:                 ; neue vektoren
5340:                 ;
5350:  81eb a2 0f     init    ldx  #$0f
5360:  81ed b5 c0     ini1    lda  lstkey,x
5370:  81ef 9d 00 49          sta  platz,x
5380:  81f2 ca                dex
5390:  81f3 10 f8             bpl  ini1
5400:                 ;
5410:  81f5 a2 01             ldx  #$01
5420:  81f7 bd 14 03  ini2    lda  $0314,x
5430:  81fa 9d 10 49          sta  platz2,x
5440:  81fd bd 03 80          lda  nvek,x
5450:  8200 9d 14 03          sta  $0314,x
5460:  8203 ca                dex
5470:  8204 10 f1             bpl  ini2
5480:                 ;
5490:  8206 60                rts
5500:                 ;
5510:                 ; restore $c0 bis $cf
5520:                 ;
5530:  8207 a2 0f     exit    ldx  #$0f
5540:  8209 bd 00 49  exi1    lda  platz,x
5550:  820c 95 c0             sta  lstkey,x
5560:  820e ca                dex
5570:  820f 10 f8             bpl  exi1
5580:                 ;
5590:  8211 a2 01             ldx  #$01
5600:  8213 bd 10 49  exi2    lda  platz2,x
5610:  8216 9d 14 03          sta  $0314,x
5620:  8219 ca                dex
5630:  821a 10 f7             bpl  exi2
5640:  821c 60                rts
5650:                 ;
5670:  821d a9 48     schlgen lda  #$48
5680:  821f 85 c2             sta  pntlh
5690:  8221 a2 47             ldx  #$47
5700:  8223 a9 a9             lda  #$a9
```



```
6340:  8282 a9 06             lda  #$06
6350:  8284 20 af 82          jsr  srg
6360:  8287 e8                inx
6370:  8288 e8                inx
6380:  8289 e0 60             cpx  #$60
6390:  828b d0 f3             bne  srg06
6400:                 ;
6410:  828d a2 05             ldx  #$05
6420:  828f bd 13 80  aga1    lda  sour2,x
6430:  8292 95 c5             sta  opc1,x
6440:  8294 ca                dex
6450:  8295 10 f8             bpl  aga1
6460:                 ;
6470:  8297 a2 51             ldx  #$51
6490:  8299 86 c2     srg00   stx  pntlh
6500:  829b a9 00             lda  #$00
6510:  829d 20 af 82          jsr  srg
6520:  82a0 e8                inx
6530:  82a1 e8                inx
6540:  82a2 e0 61             cpx  #$61
6550:  82a4 d0 f3             bne  srg00
6560:  82a6 a9 60             lda  #$60
6570:  82a8 8d ff 57          sta  $57ff
6580:  82ab 8d ff 5f          sta  $5fff
6590:                 ;
6600:  82ae 60                rts
6610:                 ;
6620:                 ; srg erzeugt pro durchlauf eine
6630:                 ; seite der scroll-routine.
6640:                 ; die speicherseite, in der die routine
6650:                 ; erzeugt wird, steht im pntlh.
6660:                 ; der acc. entscheidet, wie oft die
6670:                 ; routine scrollt. a=0 oder a=6
6680:                 ;
6700:  82af a0 00     srg     ldy  #$00
6710:  82b1 84 c1             sty  pntr1
6720:  82b3 c8                iny
6730:  82b4 91 c1             sta  (pntr1),y  ;      0 oder 6
6740:  82b6 88                dey
6750:  82b7 a9 a2             lda  #$a2      ;  ldx
6760:  82b9 91 c1             sta  (pntr1),y
6770:  82bb c8                iny
6780:  82bc c8                iny
6790:  82bd a5 c5     mal40   lda  opc1
6800:  82bf 91 c1             sta  (pntr1),y  ; ldaa oder ldaa,x
6810:  82c1 c8                iny
6820:  82c2 a5 c7             lda  s3
6830:  82c4 91 c1             sta  (pntr1),y  ; pointer lo
6840:  82c6 c8                iny
6850:                 ;
6860:  82c7 38                sec
6870:  82c8 e9 08             sbc  #$08
6880:  82ca 85 c7             sta  s3
6890:  82cc a5 c8             lda  s4
6900:  82ce 91 c1             sta  (pntr1),y  ; pointer hi
6910:  82d0 c8                iny
6920:  82d1 e9 00             sbc  #$00
6930:  82d3 85 c8             sta  s4
```

```
profi-ass 64 v2.0     seite 11

6940:  82d5 a5 c6          lda  opc2
6950:  82d7 91 c1          sta  (pntr1),y  ; staa oder staa,x
6960:  82d9 c8             iny
6970:  82da a5 c9          lda  s5
6980:  82dc 91 c1          sta  (pntr1),y  ; pointer lo
6990:  82de c8             iny
7000:  82df e9 08          sbc  #$08
7010:  82e1 85 c9          sta  s5
7020:  82e3 a5 ca          lda  s6
7030:  82e5 91 c1          sta  (pntr1),y  ; pointer hi
7040:  82e7 c8             iny
7050:  82e8 e9 00          sbc  #$00
7060:  82ea 85 ca          sta  s6
7070:  82ec c0 f2          cpy  #$f2
7080:  82ee d0 cd          bne  mal40    ; 40 mal
7090:  82f0 a9 ca          lda  #$ca
7100:  82f2 91 c1          sta  (pntr1),y  ; dex
7110:  82f4 c8             iny
7120:  82f5 a9 30          lda  #$30
7130:  82f7 91 c1          sta  (pntr1),y  ; bmi
7140:  82f9 c8             iny
7150:  82fa a9 03          lda  #$03
7160:  82fc 91 c1          sta  (pntr1),y  ;
7170:  82fe c8             iny
7180:  82ff a9 4c          lda  #$4c
7190:  8301 91 c1          sta  (pntr1),y  ; jmp
7200:  8303 c8             iny
7210:  8304 a9 02          lda  #$02
7220:  8306 91 c1          sta  (pntr1),y  ;            byte 02
7230:  8308 c8             iny
7240:  8309 a5 c2          lda  pnt1h
7250:  830b 91 c1          sta  (pntr1),y  ;     selbe seite
7260:  830d a9 ea          lda  #$ea
7270:  830f c8             iny
7280:  8310 91 c1   fill   sta  (pntr1),y nop's
7290:  8312 c8             iny
7300:  8313 d0 fb          bne  fill
7310:  8315 60             rts
7320:               ;
7330:               ; konstante
7340:               ;
7350:               ; tabelle der
7360:  8316 22 83   anftab  .wordtaste1  ; cursor rechts
7370:  8318 3f 83           .wordtaste2  ; f7
7380:  831a 24 83           .wordtaste3  ; f1
7390:  831c 31 83           .wordtaste4  ; f3
7400:  831e 3e 83           .wordtaste5  ; f5
7410:  8320 23 83           .wordtaste6  ; cursor unten
7420:               ;
7430:  8322 80      taste1  .byte$80     ; cursor rechts
7440:  8323 00      taste6  .byte$00     ; cursor unten
7450:               ;
7460:  8324 06      taste3  .byte$06        ; f1
7470:  8325 03 0a 60       .byte$03,$0a,$60,$0d
7480:  8329 60 0f 18       .byte$60,$0f,$18,$11
7490:  832d 0c 14 03       .byte$0c,$14,$03,$16
7500:               ;
7510:  8331 86      taste4  .byte$86        ; f3
```

```
profi-ass 64 v2.0     seite 12

7520:  8332 c0 0f c0       .byte$c0,$0f,$c0,$13
7530:  8336 c0 17 c0       .byte$c0,$17,$c0,$1b
7540:  833a 30 1f 30       .byte$30,$1f,$30,$23
7550:               ;
7560:  833e 00      taste5  .byte$00        ; f5
7570:               ;
7580:  833f 00      taste2  .byte$00        ; f7
7590:  8340 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7600:  8348 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7610:  8350 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7620:  8358 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7630:  8360 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7640:  8368 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7650:  8370 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7660:  8378 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7670:  8380 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7680:  8388 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7690:  8390 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7700:  8398 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7710:  83a0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7720:  83a8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7730:  83b0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7740:  83b8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7750:  83c0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7760:  83c8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7770:  83d0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7780:  83d8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7790:  83e0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7800:  83e8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7810:  83f0 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7820:  83f8 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7830:  8400 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7840:  8408 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7850:  8410 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7860:  8418 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7870:  8420 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7880:  8428 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7890:  8430 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7900:  8438 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7910:  8440 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7920:  8448 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7930:  8450 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7940:  8458 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7950:  8460 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7960:  8468 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7970:  8470 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
7980:  8478 ff ff ff        .word$ffff,$ffff,$ffff,$ffff
8000:  8480 a9 8f   txtout  lda  #<txt
8010:  8482 a0 84           ldy  #>txt
8020:  8484 20 1e ab        jsr  print
8022:  8487 a9 82           lda  #<txt2
8022:  8489 a0 85           ldy  #>txt2
8024:  848b 20 1e ab        jsr  print
8026:  848e 60              rts
8030:  848f 0e 08 93 txt    .byte14,8,147,13
8032:  8493 12 96 b4        .asc "██   RECORDER-JUSTAGE von H. Diebeck     !"
8034:  84bd b4 20 20        .asc "!  (c) INPUT 64 - Heinz Heise Verlag    !▒"
8035:  84e7 0d             .byte13
```

```
profi-ass 64 v2.0     seite 13

8040:  84e8 20 31 2e        .asc " 1. Band einlegen und starten"
8050:  8505 0d 0d           .byte13,13
8060:  8507 20 32 2e        .asc " 2. Schraube bis zum Anschlag eindrehen"
8070:  852e 0d 0d           .byte13,13
8080:  8530 20 33 2e        .asc " 3. Schraube loesen, bis im Anzeigen-   "
8090:  8558 20 20 20        .asc "    feld saubere Linien sichtbar werden"
8100:  857f 0d 0d 00        .byte13,13,0
8110:  8582 20 34 2e txt2   .asc " 4. f1 - Hilfslinien Commodore-Format   "
8120:  85aa 20 20 20        .asc "    f3 - Hilfslinien SuperTape-Format"
8130:  85cf 0d 0d           .byte13,13
8140:  85d1 20 35 2e        .asc " 5. Mit RETURN Programm verlassen"
8150:  85f2 0d 0d 00        .byte13,13,0,0
]8000-85f6
no errors
```

DER WETTBEWERB GEHT WEITER!!

3000 Mark warten auf einen neuen Gewinner. Noch einmal kurz die Bedingungen für den INPUT 64-Wettbewerb:

Sie können einsenden: Grafikprogramme, Musikprogramme, Spiele, Lernprogramme, Anwenderprogramme und natürlich völlig neue Programmideen.

Wichtig: Werfen Sie einen Blick in das Kapitel "Hinweise für Autoren", damit Ihr Programm auch innerhalb von INPUT 64 lauffähig ist.

Der Rechtsweg ist wie immer ausgeschlossen.

Nachträge und Berichtigungen

Nachtrag: Hardcopy-Funktion des INPUT 64-Betriebssystems/Ausgabe 5/85

Bei der Ausgabe 5/85 hatten leider nur die Besitzer der Diskettenversion die Möglichkeit, Hardcopies vom Textbildschirm zu erstellen. Die technischen Probleme im Zusammenspiel Kassette/Drucker, die uns kurzfristig zur Streichung der Druckoption zwangen, sind aber inzwischen beseitigt. Von dieser Ausgabe an steht die Hardcopy-Funktion wieder allen Lesern zur Verfügung.

Betrifft: Korrektur zu "DICTIONARY" im Beiheft 4/85

Leider war die Angabe, wie das Vokabelprogramm auch ohne Laden einer Datei gestartet werden kann, nicht ganz vollständig. Die komplette Anweisung muß lauten:

```
n=100:rem 100= zahl der maximal einzugebenden vokabeln

dimd$(n):dime$(n):dimdl$(50):dimel$(50):goto1800
```

ARDF/FUCHSJAGD

Zur Spielidee:

Das Graphik-/Textadventure ARDF/Fuchsjagd beruht auf dem Geländespiel Fuchsjagd der Funkamateure. Ein Spiel, das auf der ganzen Welt gespielt wird. Mehrere Sender werden in einem Gelände versteckt und müssen mit einem tragbaren Peilempfänger gefunden werden. Der 'Jäger' peilt den 'Fuchs' an. Die verschiedenen Füchse senden auf verschiedenen Frequenzen im 2-Meter Amateurfunkband. Jeder Fuchs hat zusätzlich eine eigene Kennung, die im Morse-Code gesendet wird. Die international üblichen Kennungen sind MOE, MOI, MOS, MOH und MO5.

Ist der Fuchs (Sender) gefunden, wird eine Stempelkarte zur Kontrolle mit einer am Sender befestigten Lochzange abgelocht.

Hier Ihre Aufgabe:

2m - Peilwettbewerb

Im Gelände sind drei Sender versteckt, die zu Fuß zu erreichen sind. Am dritten Sender (Kennung MOS) steht die Stempeluhr für die Zeitnahme.

Nach dem Start arbeiten die Sender nur für zwei Stunden. Ihre Laufzeit ist daher begrenzt. Im Zweifelsfall verzichten Sie also lieber auf einen Fuchs.

Die Sendefrequenzen und ihre Kennungen sind folgende:

Fuchs 1: MOE 144.136 MHz
Fuchs 2: MOI 144.096 MHz
Fuchs 3: MOS 144.056 MHz

Wir wünschen ihnen nun gutes Gelingen und viel Spaß beim Wettbewerb.
Werfen Sie ruhig öfter mal einen Blick auf die schöne Landschaft.

Hinweise zum Spiel:

Das Spiel fordert Sie mit der Frage

'WAS MACHST DU?'

zu einer Aktion auf. Ihre Antwort darf aus maximal zwei Worten bestehen, einem Verb und einem Objekt. Die Antwort muß in der 'Ich-Form' sein, zum Beispiel:

ICH UNTERSUCHE BAUM

Der Einfachheit halber wird das 'ICH' weggelassen:

UNTERSUCHE BAUM

Um sich fortzubewegen, müssen nur die Kürzel, die rechts oben stehen, eingegeben werden. Dabei bedeuten:

N=Norden, S=Süden, W=Westen, O=Osten, U=Unten

Was können Sie noch machen? Sie können zum Beispiel Gegenstände nehmen:

NEHME SCHLÜSSEL

oder auch wieder ablegen:

LEGE SCHLÜSSEL

Da Sie eine Zeitbegrenzung haben, können Sie sich die Uhr über:

ZEIT

anzeigen lassen.Wenn Sie Ihren Empfänger bedienen wollen, müssen Sie ihn vorher mit:

RX

ansprechen. Eine Frequenzeingabe machen Sie zum Beispiel so:

TIPPE 144.036

Zurück ins Spiel können Sie mit den Befehlen:

BILD, ZURÜCK oder NICHTS.

Wollen Sie Ihre letzte Eingabe noch einmal sehen geben Sie:

H

ein. Mit dem Befehl:

ENDE

können Sie das Spiel abbrechen, aber bevor Sie aus Verzweiflung aufgeben, versuchen Sie:

UNTERSUCHE (jeweiliges Objekt)

Ach ja, da gibt's noch den überaus wichtigen Befehl:

PEILE,

mit dem Sie die Füchse anpeilen können (um die geht es ja bei diesem Spiel).

Hier nun für alle Nicht-Funker einen kleinen Exkurs über's Peilen: Wenn man versucht einen Sender anzupeilen, benutzt man dazu eine richtungsempfindliche Antenne. Diese dreht man solange, bis man den gesuchten Sender möglichst stark empfängt. Nun weiß man die Richtung, in der sich der gesuchte Sender befindet, und über eine Anzeige am Peilempfänger erfährt man, mit welcher Feldstärke er zu empfangen ist. Aus der Feldstärke kann man auf die Entfernung zum Sender schliessen.

Bei diesem Spiel nimmt Ihnen der Rechner die meisten Tätigkeiten des Peilens ab. Auf die Eingabe PEILE antwortet er Ihnen mit der Himmelsrichtung, in der sich der gesuchte Sender befindet, und mit der Feldstärke, mit der er zu empfangen ist. Die Feldstärke wird mit Werten zwischen S0 bis S9 angegeben. Je größer der S-Wert, desto näher sind Sie am Fuchs. Empfangen Sie den Fuchs mit der Feldstärke S9, dann ist er in dem Bild versteckt, in dem Sie sich gerade befinden. Wenn Sie nun den Sender suchen, wird der Standort preisgegeben, und Sie können nun Ihre Karte abstempeln.

Und dann? Weiter zum nächsten Fuchs. Nehmen Sie Fuchs 3 (Kennung MOS) als letzten, weil dort auch die Stempeluhr für die Zeitnahme steht.

BASIC-Compactor und Renumber

Programmverkürzungen bis zu 30 Prozent sind durch die Kombination dieser beiden Funktionen keine Seltenheit. Dadurch wird es möglich, ausführlich kommentierte Programme zu schreiben, die man auch später noch durchschaut, und für den alltäglichen Einsatz eine komprimierte Form zu benutzen. Das spart Speicherplatz und Wartezeit.

Nach dem Laden vom eigenen Datenträger und dem Start mit RUN belegt das Programm den von BASIC nicht benutzten Adressbereich von 49152 bis 51407 ($C000 bis $C8CF). Will man ein BASIC-Programm verkürzen, wird dieses anschliessend nachgeladen. Nach dem Aufruf durch

```
SYS 50176
```

erscheint das Anfangsmenue des Compactor/Renumber-Tools:

```
bitte waehlen: compactor  c
               renumber   r
```

Empfehlenswert ist, das Programm zuerst mit möglichst niedrigen Zeilennummern und Schrittweiten neu zu numerieren. Die Zeilennummern nehmen zwar zu Anfang jeder Zeile immer zwei Bytes ein, niedrige Zeilennummern wirken sich aber bei allen Sprüngen speicherplatzsparend aus (GOTO, GOSUB usw.). So belegt das Statement GOTO50000 sechs Bytes, das Statement GOTO50 nur noch drei Bytes - jeweils ein Byte für das GOTO und eins pro Ziffer.

Ist das RENUMBER durchgeführt, müssen die anderen Funktionen über SYS 50176 erneut aufgerufen werden. Drücken Sie "c", erscheint zur Auswahl:

```
1  rem's und spaces entfernen
2  compactieren

bitte wählen Sie
```

Durch die 1.Funktion werden nur Kommentarzeilen (REM-Zeilen) und überflüssige Leerzeichen entfernt. Das Programm bleibt editierbar. Wählen Sie die "2", werden ausserdem die BASIC-Zeilen zu maximal 255 Zeichen zusammengefaßt, diese Zeilen sind anschließend nicht mehr editierbar. Sie können allerdings problemlos neue Zeilen einfügen.

Die so behandelten Files sind wie normale BASIC-Programme zu laden bzw. abzuspeichern. Es können alle reinen BASIC-Programme "compactiert" werden, mit einer Einschränkung: Die letzte Zeile darf kein REM enthalten.

Das Programm benutzt den Speicherplatz unter dem Betriebssystem (57344 bis 65535/$E000 bis $FFFF) zur Ablage von Tabellen, kollidiert also mit eventuell dort untergebrachten Spracherweiterungen oder ähnlichem.

HILFSPROGRAMME 2: Rasterzeilen-Tools

Die "Zaubereien", die mit diesem Hilfsprogramm möglich sind, haben Sie sich bestimmt schon im Demonstrationsprogramm angesehen. Lange Erklärungen zum Effekt sind deswegen wohl überflüssig. Angemerkt sei nur noch, daß wir diese Tools auch veröffentlichen als Trost für eventuelle Frustrationen aus dem letzten Teil des Video-Chip-Kurses ("in BASIC nicht programmierbar ...").

Sie können wählen zwischen fünf Möglichkeiten (siehe Tabelle 1). Eingeschaltet wird die Bildschirmteilung durch

```
SYS 52224,n,a,e
```

"n" darf Werte zwischen null und vier annehmen, die Zuordnung entspricht Tabelle 1.

TABELLE 1

- Hires-Grafik und Text....................................0
- Multicolor-Grafik und Text...............................1
- Gross-Klein-Schrift- und Gross-Grafik-Modus..............2
- zwei verschiedenen Farben für Bildschirmrahmen und -hintergrund....3
- Extented-Colour- und Normal-Modus........................4

"a" ist die Position des Elektronenstrahls, bei der der andere Modus eingeschaltet wird, "e" entsprechend die Ausschaltposition. "a" und "e" können Werte zwischen 0 und 299 annehmen. Zum Beispiel:

```
SYS 52224,0,100,200
```

schaltet bei Rasterzeilen-Position 100 die hochauflösende Grafik ein und wechselt bei Position 200 zurück in den Text-Modus. Der obere Rand des Bildschirms entspricht übrigens etwa der Position 30, nicht der Position 0.

Ist die Bildschirmteilung eingeschaltet, können die Anfangs- und Endpositionen durch

```
SYS 52669,a,e
```

verändert werden. Achtung: a und e dürfen nicht größer als 299 werden, ansonsten riskiert man einen Rechnerabsturz. Diese Fehlermöglichkeit wurde, im Gegensatz zu anderen, nicht abgefangen, da das Programm an dieser Stelle zeitkritisch ist und nicht unnütz viele Zero-Page-Adressen belegt werden sollten.

Ausgeschaltet wird der geteilte Bildschirm durch den Befehl

```
SYS 52626
```

Dieser Befehl muß eingegeben werden, um eine andere Option des Programms zu wählen. Es ist nicht möglich, nach einem SYS 52224,0,a,e

direkt zu wechseln in Multi-Color-/Text-Aufteilung durch ein SYS5224,1,a,e. Der Umweg über den Aufruf der Normalisierungsroutine ist zwingend!

Die Rasterzeilen-Tools laufen zusammen mit dem Graphik-Paket HIRESSPEED (INPUT 64 Ausgabe 4/85). Das heißt, daß beim Einschalten von Hires oder Multicolor (n=0 oder n=1) der Video-RAM nach dez. 2048 ($0800) gelegt wird, da ja der normale Textspeicher (ab dez. 1024) weiterhin zur Textdarstellung benutzt wird. Dies erfordert natürlich einen hochgesetzten BASIC-Anfang, damit sich Programmspeicher und der dann als Farbspeicher für die Grafik dienende nach dez. 2048 verlegte Video-RAM nicht "beißen". Sie müssen also den BASIC-Anfang mindestens nach dez. 3072 verlegen, durch

```
POKE 44,12:POKE (PEEK(44)*256+PEEK(43)-1),0:NEW
```

Ist dies nicht geschehen, und sie versuchen Hires oder Multicolor mit Text zu mischen, erhalten Sie eben diesen Befehl als Fehlermeldung. Bevor Sie diese POKEs ausführen, speichern Sie ein eventuell im Rechner vorhandenes BASIC-Programm ab, es wird dadurch zerstört.

Haben Sie als Wert für n drei gewählt, können in den Speicherzellen 251 und 252 die Werte für die beiden Bildschirmfarben übergeben werden. Probieren Sie zum Einstieg den Befehl

```
POKE 251,0:POKE 252,2:SYS 52224,3,100,200:
```

und ändern anschließend die Farben durch POKE 251,f1:POKE 252,f2. f1 und f2 sollten zwischen null und fünfzehn liegen, die Zuordnung von Werten und Farben können Sie dem Commodore-Handbuch entnehmen.

Das Programm wird als BASIC-File aus dem Magazin heraus abgespeichert. Nach dem Start mit RUN belegt es den Bereich von dez.52224 bis 52718 ($CC00bis $CDEE).

ZUM PRINZIP

Der Video-Chip wird so programmiert, daß der Elektronenstrahl, der den Bildschirm aufbaut, beim Erreichen einer bestimmten Position einen Interrupt auslöst. Der Interrupt-Vector ist auf eine eigene Routine gerichtet. Diese wertet aus, ob der Interrupt durch den Timer ausgelöst wurde - dann normal weiter - oder durch den VIC. Kam die Anforderung vom VIC, werden die entsprechenden Änderungen bei der Position a (oben) durchgeführt und der neue Interrupt-Punkt auf e (unten) gelegt. Löst der Elektronenstrahl an Position e einen Interrupt aus, wird in die andere Bildschirmdarstellung gewechselt und die Position a als interrupt-auslösend zurückgeschrieben. Die Grundlagen hierzu sind ausführlich beschrieben in BITS & BYTES IM VIDEO-CHIP Teil 5 (INPUT 64 Ausgabe 5/85), ebenso die Erklärung des "Extented Colour-Modus". Programmiertechnische Einzelheiten entnehmen Sie bitte nachfolgendem Listing.

```
rasterzeilen-tools by js     profi-ass 64 v2.0    seite 1

 40:   cc00                      .opt p4,oo
 50:   cc00                      *=   $cc00
 60:   cc00                      .tit "rasterzeilen-tools by js"
 62:   cc00                      .page60
 70:                    ;
 71:                    ; aufruf sys 52224,n,a,e
 72:                    ; n=0 - hires
 73:                    ; n=1 - multicolour hires
 74:                    ; n=2 - gross-klein-schrift
 75:                    ; n=3 - rahmen/hintergrund-farbe
 76:                    ; n=4 - extented colour mode
 77:                    ; org - aufruf sys 52626
 78:                    ; hol - aufruf sys 52669,a,e
 79:   0002             flag    =     02
 80:   d000             vic     =     53248    ; basisadresse video-chip
 81:   ea31             iqorg   =     $ea31   ; originaler interrupt-einsprung
 82:   0314             iqvec   =     $0314   ; zeiger auf    "
 83:   d01a             imr     =     $d01a   ; interrupt-mask-register
 84:   d019             irr     =     $d019   ; interrupt-request-register
 85:   d011             rah     =     vic+17  ; high-bit strahlposition
 86:   d012             ral     =     vic+18  ; low-bit         "
 87:   dc0d             iqfl    =     $dc0d   ; interrupt-flag
 88:   00fd             wahl    =     $fd
 89:   00fb             farbe1  =     $fb     ; dez. 251
 91:   00fc             farbe2  =     $fc     ; dez. 252
 92:   00f7             oben    =     $f7     ; beginn des vorgangs low-byte
 93:   00fe             rett    =     $fe     ; zwischenspeicher fuer vic-register
 24
 94:   00f8             an      =     $f8     ; flag fuer high-bit oben
 95:   00fa             aus     =     $fa     ;        "          unten
 96:   aefd             chkom   =     $aefd   ; prueft auf komma
 97:   b79e             getbyt  =     $b79e   ; holt wert bis 255
 98:   00f9             unten   =     $f9     ; ende des vorgangs low-byte
 99:                    ;-----------------------------------

100:  cc00 78        init     sei
100:  cc01 20 fd ae           jsr  chkom
100:  cc04 20 9e b7           jsr  getbyt
100:  cc07 86 fd              stx  wahl     ; modus holen und retten
101:  cc09 20 bd cd           jsr  hol      ; strahlpositionen holen
102:  cc0c a6 fd              ldx  wahl     ; modus zurueck nach x
104:  cc0e e0 05              cpx  #5
104:  cc10 b0 56              bcs  ill      ; kein wert > 4 !!
105:  cc12 e0 02              cpx  #2
105:  cc14 b0 12              bcs  basegal  ; kein hires oder multicolor
106:  cc16 a5 2c              lda  $2c
106:  cc18 c9 0c              cmp  #$0c
106:  cc1a b0 0c              bcs  basegal  ; basic muss bei hires ab $0c00 lieg
en
107:  cc1c a9 dc              lda  #<warntext
107:  cc1e a0 cd              ldy  #>warntext
107:  cc20 20 1e ab           jsr  $ab1e
107:  cc23 a2 80              ldx  #$80
107:  cc25 6c 00 03           jmp  ($0300)  ; ende
108:  cc28 a9 81     basegal  lda  #%10000001
108:  cc2a 8d 1a d0           sta  imr      ; interrupt durch vic zulassen
109:  cc2d 8a        i1       txa
109:  cc2e 0a                 asl
109:  cc2f aa                 tax
110:  cc30 bd 88 cd           lda  wahltab,x
110:  cc33 8d 14 03           sta  iqvec

rasterzeilen-tools by js     profi-ass 64 v2.0    seite 2

120:  cc36 bd 89 cd           lda  wahltab+1,x
120:  cc39 8d 15 03           sta  iqvec+1  ; neuen interrupt-einsprung aus tabe
lle
150:  cc3c a5 f7     write    lda  oben
150:   cc3e 8d 12 d0          sta  ral
150:   cc41 a9 00             lda  #00
150:   cc43 85 02             sta  flag
150:   cc45 ad 18 d0          lda  $d018
150:   cc48 85 fe             sta  rett
160:   cc4a ad 11 d0          lda  rah
160:   cc4d 24 f8             bit  an
160:   cc4f 10 06             bpl  inend
170:   cc51 09 80             ora  #$80
170:   cc53 8d 11 d0          sta  rah
180:   cc56 2c                .byte$2c
190:   cc57 29 7f    inend    and  #$7f
190:   cc59 8d 11 d0          sta  rah      ; strahl-position markieren
195:   cc5c a9 60             lda  #$60
195:   cc5e 8d 00 cc          sta  init     ; zweiten aufruf verhindern
197:   cc61 a9 78             lda  #$78
197:   cc63 8d 92 cd          sta  org      ; org-aufruf ermoeglichen
200:   cc66 58       inout    cli
200:   cc67 60                rts
205:   cc68 4c 48 b2 ill      jmp  $b248    ; illegal quantity error
210:                 ;-----------------------------------
300:   cc6b ad 19 d0 hires    lda  irr
310:   cc6e 8d 19 d0          sta  irr       ; irr zuruecksetzen
320:   cc71 10 41             bpl  orgcon    ; interrupt durch timer
330:   cc73 24 02             bit  flag
330:   cc75 30 16             bmi  zz       ; unten oder oben
333:   cc77 ad 18 d0          lda  $d018
333:   cc7a 85 fe             sta  rett     ; merken
335:   cc7c a9 2c             lda  #%00101100
335:   cc7e 8d 18 d0          sta  $d018    ; zeichensatz nach $2000, video-ram n
ach $0800
337:   cc81 a6 f9             ldx  unten
340:   cc83 ad 11 d0          lda  rah
340:   cc86 09 20             ora  #%00100000  ; hires-bit setzen
345:   cc88 a4 fa             ldy  aus
345:   cc8a 4c 9b cc          jmp  lowhi
350:   cc8d a5 fe    zz       lda  rett
350:   cc8f 8d 18 d0          sta  $d018    ; alte zeichensatz und video-ram posi
tion
355:   cc92 a6 f7             ldx  oben
360:   cc94 ad 11 d0          lda  rah
360:   cc97 29 1f             and  #%00011111  ; hires-bit ausblenden
361:   cc99 a4 f8             ldy  an
362:                 ;-----------------------------------
365:   cc9b 30 03    lowhi    bmi  w1       ;test auf high-bit strahl position
370:   cc9d 29 7f             and  #$7f     ; high-bit ausblenden
375:   cc9f 2c                .byte$2c      ; naechsten befehl nicht ausfuehren (
bit $8009)
380:   cca0 09 80    w1       ora  #$80     ; high-bit setzen
382:   cca2 8d 11 d0          sta  rah
385:   cca5 8e 12 d0          stx  ral      ; neuer rasterzeilenwert als interrup
tquelle
390:   cca8 a5 02             lda  flag
390:   ccaa 49 ff             eor  #$ff
390:   ccac 85 02             sta  flag     ; oben/unten-flag invertieren
490:   ccae 68       w3       pla
490:   ccaf a8                tay
490:   ccb0 68                pla
490:   ccb1 aa                tax
rasterzeilen-tools by js     profi-ass 64 v2.0    seite 3

490:   ccb2 68                pla
490:   ccb3 40                rti           ; interrupt aussprung
499:                 ;-----------------------------------
500:   ccb4 ad 0d dc orgcon   lda  iqfl     ; interruptflag loeschen
505:   ccb7 58                cli           ; interrupt zulassen
510:   ccb8 4c 31 ea          jmp  iqorg    ; und normal weiter
699:                 ;-----------------------------------
700:   ccbb ad 19 d0 grosskl  lda  irr
700:   ccbe 8d 19 d0          sta  irr
```

```
700:   ccc1 10 f1            bpl  orgcon
710:   ccc3 24 02            bit  flag
710:   ccc5 30 12            bmi  gk1
720:   ccc7 ad 18 d0         lda  $d018
720:   ccca 09 02            ora  #02
720:   cccc 8d 18 d0         sta  $d018   ;gross/klein ein
730:   cccf a6 f9            ldx  unten
730:   ccd1 ad 11 d0         lda  rah
730:   ccd4 a4 fa            ldy  aus
740:   ccd6 4c 9b cc         jmp  lowhi
800:   ccd9 ad 18 d0 gk1     lda  $d018
800:   ccdc 29 fd            and  #%11111101
800:   ccde 8d 18 d0         sta  $d018   ; gross/graphik ein
810:   cce1 a6 f7            ldx  oben
810:   cce3 ad 11 d0         lda  rah
810:   cce6 a4 f8            ldy  an
810:   cce8 4c 9b cc         jmp  lowhi
899:                    ;-----------------------------------
1300:  cceb ad 19 d0 multi   lda  irr
1310:  ccee 8d 19 d0         sta  irr
1320:  ccf1 10 c1            bpl  orgcon
1330:  ccf3 24 02            bit  flag
1330:  ccf5 30 1e            bmi  zzm
1335:  ccf7 ad 16 d0         lda  $d016
1335:  ccfa 09 10            ora  #%00010000
1335:  ccfc 8d 16 d0         sta  $d016   ; multicolor ein
1336:  ccff ad 18 d0         lda  $d018
1336:  cd02 85 fe            sta  rett
1337:  cd04 a9 2c            lda  #%00101100
1337:  cd06 8d 18 d0         sta  $d018   ; siehe hires
1338:  cd09 a6 f9            ldx  unten
1340:  cd0b ad 11 d0         lda  rah
1340:  cd0e 09 20            ora  #%00100000
1340:  cd10 a4 fa            ldy  aus
1340:  cd12 4c 9b cc         jmp  lowhi
1350:  cd15 ad 16 d0 zzm     lda  $d016
1350:  cd18 29 ef            and  #%11101111
1350:  cd1a 8d 16 d0         sta  $d016   ; multicolor aus
1355:  cd1d a5 fe            lda  rett
1355:  cd1f 8d 18 d0         sta  $d018
1357:  cd22 a6 f7            ldx  oben
1360:  cd24 ad 11 d0         lda  rah
1360:  cd27 29 1f            and  #%00011111
1360:  cd29 a4 f8            ldy  an
1360:  cd2b 4c 9b cc         jmp  lowhi
1499:                    ;-----------------------------------
1700:  cd2e ad 19 d0 farbe   lda  irr
1700:  cd31 8d 19 d0         sta  irr
1700:  cd34 30 03            bmi  faerb
rasterzeilen-tools by js     profi-ass 64 v2.0    seite 4

1700:  cd36 4c b4 cc         jmp  orgcon
1710:  cd39 24 02     faerb  bit  flag
1710:  cd3b 30 12            bmi  fb2
1720:  cd3d a5 fb     doch   lda  farbe1
1720:  cd3f 8d 20 d0         sta  $d020
1720:  cd42 8d 21 d0         sta  $d021   ; farbe1 in rahmen und hintergrund
1730:  cd45 a6 f9            ldx  unten
1730:  cd47 ad 11 d0         lda  rah
1730:  cd4a a4 fa            ldy  aus
1740:  cd4c 4c 9b cc         jmp  lowhi
1750:  cd4f
1750:  cd4f
1805:  cd4f a5 fc     fb2    lda  farbe2
1805:  cd51 8d 20 d0         sta  $d020
1805:  cd54 8d 21 d0         sta  $d021   ; farbwechsel
1810:  cd57 a6 f7            ldx  oben
1810:  cd59 ad 11 d0         lda  rah
1810:  cd5c a4 f8            ldy  an
1810:  cd5e 4c 9b cc         jmp  lowhi
```

```
1899:                    ;-----------------------------------
2300:  cd61 ad 19 d0 extend  lda  irr
2310:  cd64 8d 19 d0         sta  irr
2320:  cd67 30 03            bmi  extiq
2320:  cd69 4c b4 cc         jmp  orgcon
2330:  cd6c 24 02    extiq   bit  flag
2330:  cd6e 30 0c            bmi  zze
2335:  cd70 a6 f9            ldx  unten
2340:  cd72 ad 11 d0         lda  rah
2340:  cd75 09 40            ora  #%01000000  ; extented colour ein
2345:  cd77 a4 fa            ldy  aus
2345:  cd79 4c 9b cc         jmp  lowhi
2350:  cd7c a6 f7    zze     ldx  oben
2360:  cd7e ad 11 d0         lda  rah
2360:  cd81 29 1f            and  #%00011111  ; extented colour aus
2490:  cd83 a4 f8            ldy  an
2490:  cd85 4c 9b cc         jmp  lowhi
2499:                    ;-----------------------------------
10000: cd88 6b cc    wahltab .wordhires        ; uebergabe von 0
10003: cd8a eb cc            .wordmulti        ;    "      "  1
10005: cd8c bb cc            .wordgrosskl      ; und so weiter
10006: cd8e 2e cd            .wordfarbe
10008: cd90 61 cd            .wordextend
19000:                   ;--------------------------------
20000: cd92 60       org     rts          ; wird zu sei nach aufruf von init
20002: cd93 a9 00            lda  #00
20002: cd95 8d 1a d0         sta  imr     ; imr loeschen
20005: cd98 a9 ea            lda  #$ea
20005: cd9a 8d 3b 01         sta  0315    ; und
20010: cd9d a9 31            lda  #$31
20010: cd9f 8d 3a 01         sta  0314    ; interrupt-vector auf original-routi
ne
20012: cda2 a9 1b            lda  #27
20012: cda4 8d 11 d0         sta  $d011   ; hires und extented colour aus
20015: cda7 a9 c8            lda  #200
20015: cda9 8d 16 d0         sta  $d016   ; multicolor aus
-20020: cdac a5 fe           lda  rett
20040: cdae 8d 18 d0         sta  $d018   ; wert zurueckschreiben
20050: cdb1 a9 60            lda  #$60
20050: cdb3 8d 92 cd         sta  org

rasterzeilen-tools by js     profi-ass 64 v2.0    seite 5

20050: cdb6 a9 78            lda  #$78
20050: cdb8 8d 00 cc         sta  init    ; richtige reihenfolge !!
20100: cdbb 58       oend    cli
20100: cdbc 60               rts
29000:                   ;--------------------------------
30142: cdbd 20 cc cd hol     jsr  hol2
30142: cdc0 86 f7            stx  oben
30142: cdc2 85 f8            sta  an      ; obere und
30144: cdc4 20 cc cd         jsr  hol2
30144: cdc7 86 f9            stx  unten
30144: cdc9 85 fa            sta  aus     ; untere position holen
30145: cdcb 60               rts
30146: cdcc 20 fd ae hol2    jsr  chkom
30146: cdcf 20 8a ad         jsr  $ad8a
30146: cdd2 20 f7 b7         jsr  $b7f7   ; adresse  holen und nach $14/$15
30148: cdd5 a6 14            ldx  $14
30148: cdd7 a5 15            lda  $15
30148: cdd9 6a               ror
30148: cdda 6a               ror          ; wenn 1, dann 128, sonst 0
30200: cddb 60               rts
35000:                   ;--------------------------------
35010: cddc 0d       warntext .byte13
35020: cddd 50 4f 4b          .asc "poke 44,12:poke(peek(44)*256+peek(43)-1),0:n
ew:"
35030: ce0c 20 00             .byte32,0
]cc00-ce0e
no errors
```

NEWS

In unserer Neuigkeiten-Vorstellung geht es diesmal um ein neues ROM-Modul für den C 64 und einen - hoffentlich - haltbaren Joy-Stick mit Micro-Schaltern. Hier noch einmal die Bezugsadressen:

Competition PRO 5000:	Dynamics marketing Gmbh Große Bäckerstraße 11 2 Hamburg 11 Preis: 72.- DM.
Business Basic V6.5	King Soft Fritz Schäfer Schnackebusch 4 5106 Roetgen Preis: 199.- DM.

Übrigens: Die "News" sind, wie auch andere Teile des Magazins, nicht auf eigenen Datenträger "SAVE"bar.

64er-TIPS

Diesmal werden wir Sie vielleicht verunsichern; oder haben Sie schon gewußt, daß Sie Ihrem C 64 nicht alle (Rechen-)Ergebnisse glauben dürfen?

Bei welchen Aufgabenstellungen der BASIC-Interpreter ins "Schleudern" kommt, und wie Sie dieses eventuell ausgleichen können, erfahren Sie in dieser Ausgabe.

Außerdem erläutern wir Ihnen, wie Zahlen im Rechner verschlüsselt und "abgelegt" werden.

HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise:
Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:
Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter.
Das Programm darf nur in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden.
Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein.
Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht.
Senden Sie uns Ihre Programme bitte auf Kassette oder Diskette mit kommentiertem Listing und Beschreibung.
Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlichtlich erschwert wurde, zukünftig ungeprüft zurücksenden.
Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.
Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse:

1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0800-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF).

2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein.

3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein.

4. Die CTRL-Taste darf nicht benutzt werden.

Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

HIRESSPEED in Maschinensprache

Das Programm HIRESSPEED (INPUT 64 4/85) ist auch von Maschinensprache aus zu verwalten. Aufgrund einiger Anfragen die zu beachtenden Regeln und Adressen:

Es muß dafür gesorgt werden, daß die Koordinaten der Punkte keine unzulässigen Werte annehmen, da Fehler nicht mehr abgefragt werden. Fast alle Unterprogramme belegen die Adressen 247 bis 254 und verändern die Prozessor-Register. Beim FILL-Befehl wird der Cassetten-Puffer als Ablage für die Tabelle genutzt, beim TEXT-Befehl die Adressen $22/$23 als Zeiger.

Der Modus ist in Adresse $0C48 festgelegt. Steht dort eine Null, wird gesetzt, sonst wird gelöscht. Dies gilt nicht für den FILL-Befehl!

Hires einschalten	$0DDB		
Hires ausschalten	$0DF9		
Hires löschen	$0D73		
Farben setzen	$0D5A	Farbcode (zf*16+hf) im Akku	
Punkt setzen	$0E33	x-low	$0C4B
		x-high	$0C4C
		y	$0C4D
		-- beim Einsprung ab $0E1C werden	
		-- falsche Koordinaten ignoriert, es	
		-- kommt nicht zum Absturz	
Linie zeichnen	$0FDF	x-Anfang-low	$0C58
		x-Anfang-high	$0C59
		y-Anfang	$0C5A
		x-Ende-low	$0C5B
		x-Ende-high	$0C5C
		y-Ende	$0C5D
Kreis zeichnen	$0EE7	x-low	$0C4F
		x-high	$0C50
		y	$0C4E
		x-Radius	$0C51
		y-Radius	$0C52
		-- zu den Radien muß jeweils eins ad-	
		-- diert werden, die Werte dürfen 129	
		-- nicht übersteigen.	
FILL	$119B	x-low	y-Register
		x-high	Akku
		y	x-Register
TEXT	$150F	Länge des Strings im Akku	
		Zeiger auf String in $22/$23	
		Schriftbreite in $0C71	
		x-low	$0C4F
		x-high	$0C50
		y	$0C4E
GSAVE	$1774	die Fileparameter müssen gesetzt sein	
GLOAD	$178B	siehe GSAVE	

BEI LADEPROBLEMEN:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft.

Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben.

Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

TONKOPF SELBST JUSTIEREN

Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/Wiedergabekopf justieren können.

Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kassettenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage.

Legen Sie nun die zweite Seite von INPUT-64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung "PRESS PLAY ON TAPE" kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar - bei nicht total verstellter Spurlage - völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber

Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am besten markieren), und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenen Vermerk an den Verlag ein (Adresse siehe Impressum).

PS: Ist Ihr Recorder erst richtig justiert, schauen Sie sich einmal das Programm RECORDER-JUSTAGE in dieser Ausgabe an. Damit sind die Probleme beim Lesen von Kassetten - und zwar ganz gleich, welcher Herkunft und welchen Aufzeichnungsformats - endgültig behoben!

listing justage

```
800 fori=49199to49410:readd:ps=ps+d:pokei,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys12*16↑3+11*16+10
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

Für Nachzügler und "Spätzünder"

Wegen der großen Nachfrage haben wir bereits vergriffene Ausgaben von INPUT 64 nachproduziert, so daß alle bisher erschienenen Ausgaben wieder lieferbar sind!! Ab Ausgabe 4/85 ist INPUT 64 auch auf Diskette erhältlich.

Richten Sie Ihre Bestellung bitte direkt an den Verlag:

INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorkasse. Fügen Sie Ihrer Bestellung deshalb bitte einen Vorrechnungsscheck oder einen von Ihrer Bank gezeichneten Einzahlungsbeleg bei.

ÜBRIGENS: Abonnenten sparen den Preis von zwei INPUT 64 - Ausgaben pro Jahr. Und zwar ohne den üblichen 'Pferdefuß', denn das INPUT 64 - Abo ist jederzeit kündbar!

Die Kassettenversion kostet DM 12.80 , die Diskettenversion DM 19.80 je Ausgabe (einschließlich Porto und Verpackung).

INPUT 64 Ausgabe 1/85



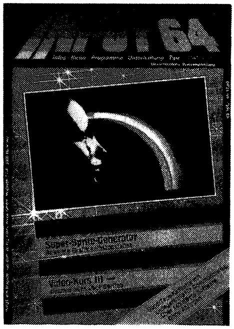
* Musikprogrammierung: SOUNDCONTROL*
HILFSPROGRAMME: DATA-Generator, LIST im Programm, RE-NEW, BASIC 2-mal im RAM *
Spiele: BALLONJAGD, INFERNO, TOWER *
Tools: MASKENGENERATOR, SPLITSCREEN *
Verwaltungsprogramm: DATEI-KASTEN *
Kurs: BITS & BYTES IM VIDEO-CHIP Teil 1
* NEWS * INTERNES *KONTAKT-ECKE *
Serie: 64er - TIPS *

INPUT 64 Ausgabe 2/85



* Textverarbeitung: SCRIPTOR * Basteltip: RESET-Taster * Spiele: LABYRINTH, SUPERSONIC * HILFSPROGRAMME: REM-OUT, DELETE, MERGE, DIRECTORY, RENUMBER * Kurs: BITS & BYTES IM VIDEO-CHIP Teil 2 * NEWS *Serie: 64er - TIPS * Tool: ZEICHENGENERATOR * Vokabelprogramm: DICTIONARY * KONTAKTECKE * LAST NOT LEAST

INPUT 64 Ausgabe 3/85



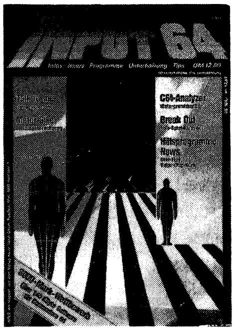
* Spriteeditor: MOBED * Kurs: BITS & BYTES IM VIDEO-CHIP Teil 3 * Maschinensprachemonitor: MLM 64 * HILFSPROGRAMME: PRINT AT, Fehlerkanal auslesen, KALT, sieben SPRITE - Befehle * Spiele: BIATHLON, HELLS BELLS * Serie: 64er - TIPS * Anwendung: AUTOSTART-Generator, FORMEL-PLOTTER * NEWS * KONTAKTECKE * LAST NOT LEAST*

INPUT 64 Ausgabe 4/85



* Graphikpaket: HIRESSPEED * Schnellladesystem: SUPERTAPE - D II * Kurs: BITS & BYTES IM VIDEO-CHIP Teil 4 * HILFSPROGRAMM: Kassetten-Directory * Spiele: REVERSI, ARTEMIS * Serie: 64er - TIPS * Anwendung: KALENDER * KONTAKTECKE * NEWS * SHORTSAVE * LAST NOT LEAST *

INPUT 64 Ausgabe 5/85



* Hintergrundmonitor: C-64 ANALYZER * Dialog mit dem Rechner: TALK TO ME * Lernprogramm: MATHE MIT NICO Teil 1* Kurs: BITS & BYTES IM VIDEO-CHIP letzter Teil * Anwendung: MOTORBIKE * NEWS *Videospiel-Klassiker: BREAK OUT * HILFS-PROGRAMME: AUTO, FIND, DUMP * Serie: 64er - TIPS * RAM - FLOPPY * LAST NOT LEAST *

Ab 15. Juli an Ihrem Kiosk: INPUT 64 Ausgabe 7/85

Wir bringen unter anderem:

- TAPE-COPY
  Endlich ein universales Kopiersystem für Kassettenbesitzer. Kopiert Commodore-Files in's SuperTape-Format, SuperTape nach Commodore, Commodore nach Commodore und SuperTape nach SuperTape.

- BOXEN-BERECHNUNG
  Schluß mit den Experimenten! Dieses Programm berechnet passend für alle gängigen Lautsprechertypen das optimale Gehäuseformat.

- und außerdem:
  Mathe mit Nico/Teil 3 * 64er-Tips - Programmieren für Einsteiger: Read, Data, Poke * Serie: SID-Kurs - Das erste Musikprogramm * Hilfsprogramme * News * und so weiter

---

c't-Magazin für Computertechnik

c't 7/85 - jetzt am Kiosk

Nicht nur für Anfänger: Neue Serie "Einstieg in CP/M" * IEC-Bus-Interface für C 64 * Video-Digitizer für C 64 im Test * Einführung in LISP * Okimate 20 auf dem Prüfstand * SuperTape für MZ-700 * Wie schnell ist das CPC-464-BASIC? * Applikation: PIO 8255 * Sound und Farbe auf dem C 16 u.v.a.m

---

elrad-Magazin für Elektronik

elrad-Doppelheft Sommer '85 - ab 1.7.85 am Kiosk

Bauanleitungen: Effektivwert-Millivoltmeter * Video-Effekt-Gerät * Klirrfaktormesser * Geiger-Müller-Zähler * Schaltungssammlung: IC-Magazin * Bühne/Studio: Curtis-ICs * Report: Glasfaser

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 1
Tel.: (05 11) 53 52-0

Technische Anfragen
nur dienstags von 9-16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Köhler & Lippmann, Braunschweig.

Druck:
Leunisman GmbH, Hannover
Hahn-Druckerei, Hannover

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198,-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

ISSN 0177-3771

Titelidee: **INPUT 64**
Titelfoto: Artreferenz

Titelmusik: **INPUT 64**

hier abtrennen

INPUT 64-Abonnement Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat ____________

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.

○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift
Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

____
Vorname/Zuname

____
Beruf/Funktion

____
Straße/Nr.

____
PLZ Wohnort

____
Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift ____________
Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

____
Name des Kontoinhabers

____
Bankleitzahl Konto-Nr.

____
Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT 64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1